KOWAL

MICRO APPLICATION

13

AMSTRAD

DES IDÉES POUR LES CPC



UN LIVRE DATA BECKER

KOWAL

MICRO APPLICATION

13

AMSTRAD

DES IDÉES POUR LES CPC



UN LIVRE DATA BECKER

Distribué par : MICRO APPLICATION
13, Rue Sainte Cécile
75009 PARIS

et

EDITION RADIO
3, Rue de l'Eperon
75006 PARIS

ISBN : 2-86899-031-2

Traduction Française assurée par Pascal HAUSMANN

Table des matières

ATTENTION

- Le caractère "^" que l'on rencontre dans certains listings correspond au symbole mathématique "puissance". On l'obtient en tapant la touche à gauche de "CLR".

- Les caractères @ \ [ ] | { } se trouvant dans les chaines alphanumériques de certains listings correspondent aux accents français. Ils apparaissent normalement à l'écran et à l'imprimante lorsque les programmes dans lesquels ils se trouvent ont été exécutés.

0. Préface

La baisse de prix considérable qui s'est produite sur le marché des ordinateurs a fait qu'on peut aujourd'hui acheter des microordinateurs pour relativement peu d'argent alors qu'ils étaient encore inabordables il y a seulement dix ans.
Alors qu'autrefois seules de grandes structures telles que les ministères, les universités ou les grandes entreprises étaient à même d'utiliser des ordinateurs pour résoudre certains problèmes déterminés, nous trouvons aujourd'hui déjà chez de nombreuses personnes privées des machines électroniques qui peuvent être comparées de par leur puissance à un petit centre de calcul d'autrefois.

Tout acquéreur d'un microordinateur (avec ou sans périphérie conséquente) finit par se demander, une fois qu'il a essayé quelques jeux électroniques, quelles tâches son ordinateur peut effectuer.

Cette question, ainsi que l'explication de toutes les possibilités et particularités de l'AMSTRAD CPC, figure au coeur de cet ouvrage. Il ne s'agit cependant pas de donner au lecteur des solutions toutes prêtes (il pourrait dans ce cas demander directement à son revendeur en microinformatique des logiciels tout prêts). Il s'agit bien plutôt de montrer au lecteur grâce aux descriptions de programmes comment des idées peuvent être formulées en problèmes et converties en programmes.

Les différents programmes sont écrits en BASIC, langage qui joue bien sûr aujourd'hui un rôle dominant dans le monde de la microinformatique. L'étude de ce livre suppose que le lecteur possède un certain nombre de connaissances de base sur le BASIC et qu'il ait étudié de façon suffisamment approfondie le manuel d'utilisation de l'AMSTRAD CPC pour pouvoir comprendre les différents programmes proposés ici, même si nous reconnaissons volontiers que ce n'est pas toujours très facile.

C'est pourquoi les programmes ont été construits de façon à ce qu'on puisse facilement comprendre ce qui s'y passe. Ce principe a bien sûr nuit à l'élégance de programmation et à la vitesse de calcul. Cet ouvrage n'atteint donc pleinement son but que lorsque le lecteur se saisit des programmes présentés pour les modifier ou les adapter à ses propres besoins. Si vous rencontrez ce faisant des obstacles qui vous semblent insurmontables, rappelez-vous que:

c'est en forgeant qu'on devient forgeron

Les différents chapitres de cet ouvrage sont en règle général organisés d'après le schéma suivant:

a) présentation d'une idée et formulation en problème

b) programme

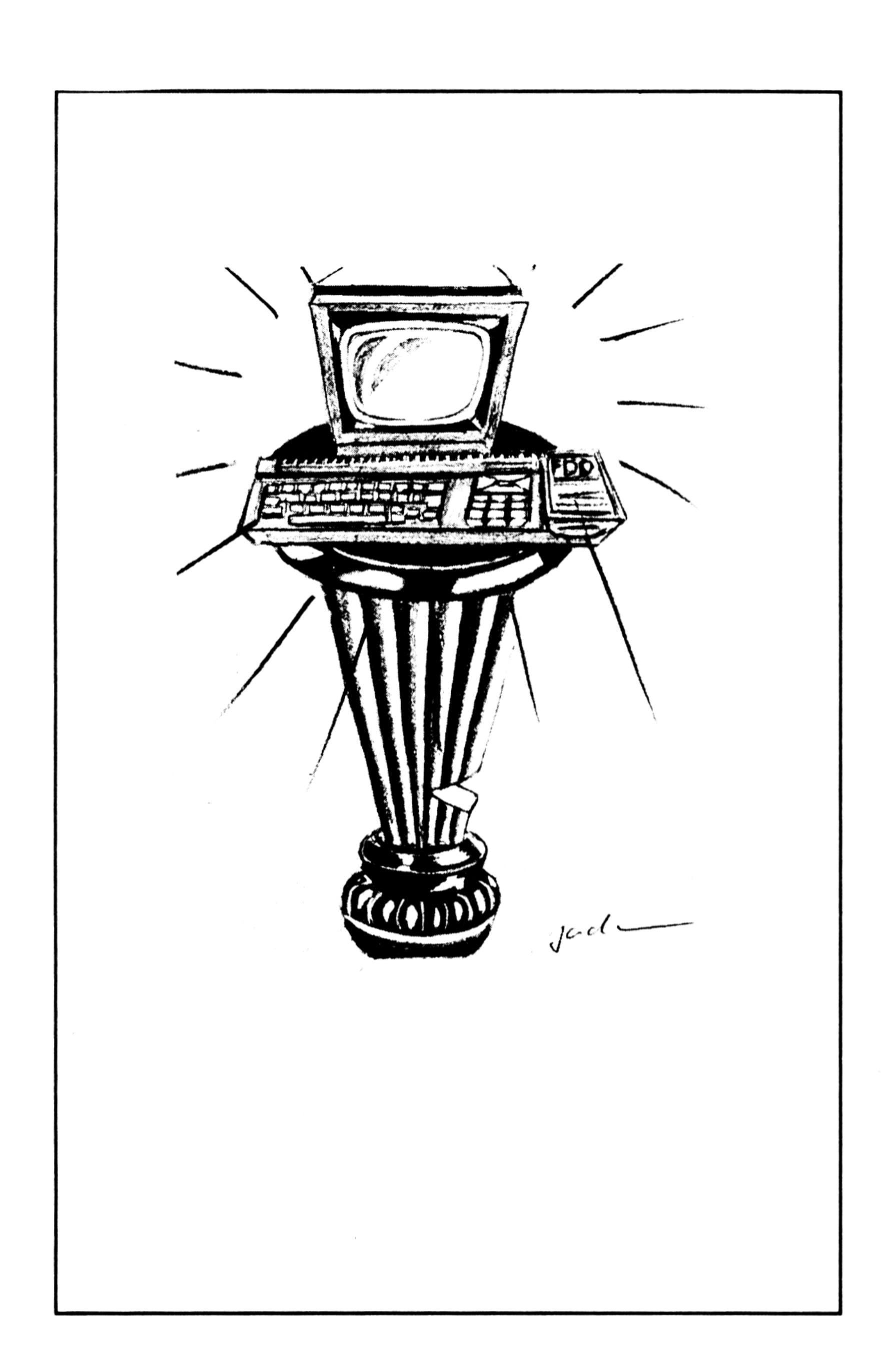
c) liste des variables utilisées dans le programme

d) description du programme

et le cas échéant

e) résultats du programme

Les idées évoquées entraînent des solutions de technique de programmation de niveau plus ou moins difficile. Nous avons cependant veillé à ce que les idées ou propositions soient présentées au début d'un nouveau chapitre avec des solutions de programme plus simples. Le lecteur qui n'est pas encore un expert en BASIC pourra ainsi aborder progressivement des structures de programme plus compliquées. Il faut d'autre part noter d'une manière générale que les entrées absurdes ou impossibles par rapport à la logique du programme ne sont pas toujours interceptées par nos programmes. Mais cela sera signalé pour chaque cas.



1. Le CPC et ce qu'il permet de faire

Sur le marché des microordinateurs on peut trouver maintenant toute une série d'ordinateurs comparables dans la même classe de prix. Il n'arrive que rarement qu'une machine électronique de ce type se distingue vraiment des autres. C'est toutefois le cas de l'AMSTRAD (sous quelque version que ce soit) car il présente quelques particularités remarquables inégalées.

Il s'agit notamment de caractéristiques musicales et graphiques extraordinaires, de la possibilité de définir des fenêtres (windows) ou de programmer des interruptions pour exécuter plusieurs tâches en même temps ou pour programmer dans le temps l'exécution de ces différentes tâches. Nous expliquerons plus en détail ces particularités lorsque nous aurons à les utiliser dans les chapitres suivants.

Dans sa configuration de base, le système AMSTRAD CPC est déjà une unité informatique prête à fonctionner, qui permet d'exécuter les étapes de base du traitement de données:

- Entrée des données ou informations (au clavier)

- Stockage des données ou informations (interne ou externe)

- Traitement ou évaluation de ces données

- Sortie des résultats (à l'écran)

Un inconvénient de l'utilisation d'un lecteur de cassette comme moyen de stockage externe des données est qu'on ne peut écrire ou lire les cassettes que de façon séquentielle. S'il s'agit par exemple de lire sur une cassette un fichier déterminé, cette cassette doit tout d'abord être embobinée jusqu'à l'endroit voulu. Cela peut entraîner dans certains cas un travail de recherche très fastidieux.

Il est de toute façon recommandé de ne pas stocker trop d'informations sur une face de cassette et de noter les emplacements des différents fichiers grâce au numéro de compteur.

L'utilisation d'un lecteur de disquette est certainement plus pratique. La disquette offre en effet une capacité de stockage plus importante et permet un accès très rapide à des emplacements déterminés (la disquette est en rotation permanente et très rapide dans le lecteur de disquette). C'est pourquoi on parle d'une mémoire avec accès direct. En ce qui concerne les instructions BASIC à utiliser, le lecteur de disquette ne se distingue pas du lecteur de cassette. Les programmes sont ainsi entièrement compatibles.

L'utilité d'un ordinateur n'est effective qu'à partir du moment où l'utilisateur dispose de logiciels appropriés. Il y a trois possibilités d'obtenir de bons programmes correspondant à des besoins individuels:

1) Acheter des logiciels commerciaux
2) Taper des listings de programmes
3) Développer soi-même des programmes

Si vous manquez de temps ou d'expérience pour programmer vous-même, vous en êtes réduit à vous reporter au marché des logiciels. C'est notamment le cas si vous voulez utiliser l'ordinateur que vous venez d'acheter directement de façon plus ou moins professionnelle (par exemple pour du traitement de texte, de la comptabilité ou du calcul). Le choix en programmes commerciaux est cependant très grand. Pour le CPC sont notamment apparus dernièrement, sur le marché, des programmes de prix et qualité très variables. Avant d'acheter un logiciel, il importe en tout cas d'effectuer des comparaisons très précises en prenant absolument en compte les points suivants:

- Qualité du manuel d'utilisation
- Services du revendeur
- Application particulière pour l'utilisateur
- Tests de logiciels dans les magazines spécialisés

Taper des listings de programmes n'est pas sans poser de petits problèmes. Suivant le sérieux des éditeurs on peut en effet s'attendre à ce que les listings contiennent plus ou moins d'erreurs. Comme il est très difficile de pénétrer dans la logique d'un programme inconnu si ce programme n'est pas accompagné d'explications suffisantes, la recherche et la correction des erreurs risquent de prendre plus de temps que le développement d'un programme original. Avant de taper un long listing de programme, il est donc recommandé de vérifier la source de ce programme.

De nombreux détracteurs de l'usage privé des ordinateurs prétendent qu'il ne s'agit pas là d'une activité créatrice. En effet, ne faire "que" jouer avec un ordinateur peut devenir à la longue une activité assez stupide. Mais la microinformatique devient un hobby très créatif à partir du moment où on écrit ou adapte soi-même des programmes. Cet ouvrage a justement été conçu pour ceux qui souhaitent puiser à la bonne source de nouvelles idées pour utiliser leur ordinateur de façon créative. C'est pourquoi cet ouvrage aborde les sujets et les suggestions les plus diverses pour les transformer en programmes. A ces programmes tout prêts (mais qui peuvent bien sûr être modifiés ou adaptés) viennent s'ajouter d'autres propositions pour chaque groupe d'idées, regroupées dans le chapitre "Le puits à idées". Etant donné la diversité des thèmes abordés, chacun devrait pouvoir y trouver son bonheur.

2. De petites choses utiles et faciles à programmer

2.1 Remarque préalable

Nous présenterons dans ce premier chapitre d'idées des solutions de programmes simples et plus aisées à assimiler. Les explications de programme seront en outre très complètes, surtout compte tenu du faible degré de difficulté des programmes. Nous espérons que les lecteurs inexpérimentés trouveront ainsi dans ce chapitre une bonne introduction aux programmes un peu plus complexes des chapitres suivants.

Il s'agit quant au fond, notamment pour les trois premiers programmes, de solutions de programme pouvant être utilisées comme sous-programmes dans des programmes plus complets et plus longs. C'est ainsi que vous rencontrerez par exemple fréquemment dans la suite de cet ouvrage le calcul de pourcentage qui peut se révéler très utile dans la solution de certains petits problèmes.

2.2 Le calcul de pourcentage

Il est souvent nécessaire de comparer entre elles plusieurs valeurs. Nous utilisons à cet effet le nombre 100 comme base de comparaison.

Les concepts de base suivants pour le calcul de pourcentage doivent être d'abord définis:

1. Valeur de base:

   La valeur de base correspond au tout, soit à 100 centièmes ou 100 %. Il s'agit toujours d'un nombre suivi d'une dénomination (F, kg, m ou autres).

2. Valeur de pourcentage:

   La valeur de pourcentage est une fraction de la valeur de base. Il s'agit également d'un nombre suivi d'une dénomination.

3. Pourcentage:

   Le pourcentage est également une fraction de la valeur de base mais fournie par rapport à 100. Le pourcentage correspond à la valeur de pourcentage.

Un petit exemple vous facilitera le maniement de ces concepts:

Supposons qu'un logiciel coûte 898 F chez le revendeur xyz. Vous allez maintenant chez un autre revendeur qui veut rester 4% (100 - pourcentage) en dessous de ce prix. Soit vous demandez à ce revendeur le prix exact, soit vous le calculez vous-même avec la formule suivante:

$$\text{Valeur de pourcentage} = \frac{\text{Valeur de base * pourcentage}}{100}$$

$$= \frac{898\ F * 96}{100} = 862{,}08\ F$$

Vous pensez certainement que l'utilisation d'un ordinateur pour des calculs aussi simples ne s'impose pas. Mais vous ne faites probablement pas de tels calculs de tête. Il vous faut certainement une calculatrice ou au moins du papier et un crayon. Et peut-être vous faut-il réfléchir un petit peu pour retrouver la formule de calcul d'une valeur de pourcentage. Le programme suivant vous permet en outre également, si vous le souhaitez, de calculer la valeur de base ou le pourcentage. Tout le monde n'a pas en tête les formules de calcul correspondantes, bien que celles-ci soient très simples:

$$\text{Pourcentage} = \frac{\text{Valeur de pourcentage * 100}}{\text{Valeur de base}}$$

$$\text{Valeur de base} = \frac{\text{Valeur de pourcentage * 100}}{\text{Pourcentage}}$$

Programme:

```
10 REM I1
20 CLS
30 PRINT"I1 - Programme de calcul de"
40 PRINT TAB(6)"valeurs de pourcentage, valeurs"
50 PRINT TAB(6)"de base ou pourcentages"
60 PRINT:PRINT:PRINT
70 PRINT TAB(11)"Bernd Kowal, 1985"
80 GOSUB 380
90 PRINT TAB(16)"Selection":PRINT:PRINT:PRINT:PRINT
100 PRINT"Valeur de pourcentage =          1":PRINT
110 PRINT"Valeur de base        =          2":PRINT
120 PRINT"Pourcentage           =          3":PRINT
130 PRINT:PRINT:PRINT
140 PRINT"Veuillez entrer le nombre correspondant"
150 PRINT"a la valeur inconnue."
160 PRINT:INPUT"Votre choix ";a
170 IF a<1 OR a>3 THEN 220
180 CLS
190 PRINT"Veuillez entrer les valeurs"
200 PRINT"correspondant aux grandeurs connues.":PRINT:PR
INT
210 ON a GOSUB 230,280,330
220 PRINT:PRINT:PRINT:PRINT"Fin des calculs":END
230 INPUT"Valeur de base ";gw:PRINT
240 INPUT"Pourcentage ";ps
250 pw=gw*ps/100
260 PRINT:PRINT:PRINT"La valeur de pourcentage est de :"
;pw
270 RETURN
280 INPUT"Valeur de pourcentage ";pw:PRINT
290 INPUT"Pourcentage ";ps
300 gw=pw*100/ps
310 PRINT:PRINT:PRINT"La valeur de base est de :";gw
320 RETURN
330 INPUT"Valeur de pourcentage ";pw:PRINT
340 INPUT"Valeur de base        ";gw
350 ps=pw*100/gw
360 PRINT:PRINT:PRINT"Le pourcentage est de :";ps
370 RETURN
380 REM SP Attendre
390 LOCATE 7,25
400 PRINT"Frappez une touche S.V.P"
410 x$=INKEY$:IF x$="" THEN 410
420 CLS:RETURN
```

Liste de variables:

a = variable de réponse pour le menu de sélection

gw = valeur de base

ps = pourcentage

pw = valeur de pourcentage

x$ = variable auxiliaire restituant la valeur de la touche appuyée

Comme la variable x$ a la même fonction dans tous les programmes de cet ouvrage, nous ne la présenterons plus dans les chapitres suivants.

Description du programme:

Ligne 10 : Commentaire (n'est pas sorti)

Ligne 20 : Vidage de l'écran

Lignes 30- 70 : Sortie du titre

Ligne 80 : Appel du sous-programme "attendre"

Lignes 90- 150 : Sortie du menu de sélection

Ligne 160 : Une valeur correspondant au menu de sélection est réclamée

Ligne 170 : On saute à la fin du programme si une valeur trop grande ou trop petite a été entrée

Lignes 180-200 : L'écran est vidé et un commentaire concernant les entrées encore à faire est sorti

Ligne 210 : Suivant le point du menu sélectionné, on saute au sous-programme correspondant

Ligne 220 : Fin du programme

Lignes 230-270 : Sous-programme de calcul de la valeur de pourcentage
- 230-240 : Les valeurs connues, valeur de base et pourcentage sont demandées
- 250 : Calcul de la valeur de pourcentage
- 260 : Sortie du résultat
- 270 : Retour du sous-programme

Lignes 280-320 : Sous-programme de calcul de la valeur de base. La procédure correspond à celle des lignes 230-270.

Lignes 330-370 : Sous-programme de calcul du pourcentage. On procède comme pour les deux sous-programmes précédents.

Lignes 380-420 : Sous-programme d'attente
- 380 : Commentaire
- 390 : Le curseur est amené sur l'emplacement indiqué
- 400 : Sortie d'un commentaire
- 410 : Interruption du déroulement du programme jusqu'à ce qu'on appuie sur une touche
- 420 : L'écran est vidé et il y a retour du sous-programme

Résultats du programme:

Si nous lançons le programme avec l'instruction RUN, le titre du programme apparaît à l'écran. Si nous appuyons sur une touche, le menu est sorti et une entrée est demandée pour la valeur inconnue. Si nous entrons par exemple la valeur 1, un commentaire est sorti et un nombre est réclamé comme valeur de base. Si nous entrons par exemple la valeur 898, une valeur pour le pourcentage sera encore demandée. Après que vous ayez entré le nombre 96, l'ordinateur vous répond:

La valeur de pourcentage est de : 862.08

Fin des calculs
Ready

Rappelons que le programme a été conçu de façon suffisamment générale pour calculer pour n'importe quelles entrées la valeur de pourcentage, la valeur de base ou le pourcentage. Les problèmes soulevés par cette proposition de programme ne nécessitent pas en fait une description aussi détaillée que celle que nous vous donnons ici. Il s'agit surtout de vous montrer le principe qui sera suivi dans les différents chapitres. Nous commenterons donc par la suite de façon plus brève les programmes simples.

2.3 Minimum, maximum et moyenne

Supposez que vous soyez le président d'une association et que vous vous intéressiez à la structure d'âge des membres de l'association. Quoi de plus naturel que de rechercher dans ce cas l'âge du plus jeune, du plus âgé ainsi que l'âge moyen (il s'agit ici de la moyenne arithmétique) des membres de l'association.

Alors qu'il vous fallait auparavant du papier, un crayon et un certain temps pour répondre à ces questions, il vous suffit aujourd'hui d'employer votre CPC. Il vous suffit d'avoir le programme approprié et vous le trouverez justement à la page suivante.

Naturellement, ce programme ne marche pas seulement pour les indications d'âge mais également pour n'importe quels nombres. Vous pouvez donc l'utiliser même si vous n'êtes pas président d'une association.

Si vous disposez d'une masse importante de données, il peut être plus judicieux de ne pas entrer les différentes valeurs avec l'instruction INPUT mais avec les instructions DATA et READ (voir chapitre 2.4) car cette seconde méthode permet plus aisément de corriger les fautes de frappe qui peuvent aisément survenir dès que vous avez de nombreuses données à entrer.

Programme:

```
10 REM I2
20 CLS
30 PRINT"I2 - Programme de recherche des"
40 PRINT TAB(6)"maximum et minimum d'un bloc de"
50 PRINT TAB(6)"donnees, avec calcul de la moyenne"
60 PRINT
70 PRINT
80 PRINT
90 PRINT TAB(11)"Bernd Kowal, 1985"
100 GOSUB 280
110 INPUT"Combien de donnees ";n
120 IF n=0 THEN 270
130 DIM a(n)
140 PRINT:PRINT"Entree des donnees :":PRINT
150 FOR i=1 TO n
160 PRINT "Valeur ";i;:INPUT a(i)
170 s=s+a(i)
180 NEXT i:CLS
190 d=s/n:k=a(1):g=a(1)
200 FOR i=2 TO n
210 k=MIN(k,a(i))
220 g=MAX(g,a(i))
230 NEXT i
240 PRINT:PRINT"Minimum                    =";k
250 PRINT:PRINT"Maximum                    =";g
260 PRINT:PRINT"Moyenne                    =";d
270 PRINT:PRINT:PRINT"Fin du Programme":END
280 REM SP attendre
290 LOCATE 7,25
300 PRINT"Veuillez frapper une touche"
310 x$=INKEY$:IF x$="" THEN 310
320 CLS:RETURN
```

Liste de variables:

a(i) = les valeurs entrées

d = valeur moyenne

g = maximum

i = index de comptage

k = minimum

n = nombre des valeurs entrées

s = somme des valeurs entrées

Description du programme:

Lignes 10-100 : Titre

Ligne 110 : Une valeur pour le nombre de nombres à entrer est demandée

Ligne 120 : On saute à la fin du programme si aucun nombre ne doit être entré

Ligne 130 : On réserve de la place en mémoire pour le tableau a(n)

Lignes 140-180 : Entrée des données. Les différentes valeurs sont additionnées directement dans le tableau s

Ligne 190 : La valeur moyenne est calculée et on suppose comme grandeur de départ pour les variables k et g la valeur entrée en premier, a(1)

Lignes 200-230 : Toutes les valeurs entrées sont comparées avec la plus petite (ligne 210) et la plus grande (ligne 220) valeurs. Si une valeur est inférieure au minimum précédent ou supérieure au maximum précédent, les valeurs des variables k ou g sont modifiées en conséquence. La boucle de programme commence d'abord par la valeur i=2 puisque les variables k et g se sont vu affecter en ligne 190 la première valeur a(1).

Lignes 240-260 : Sortie du résultat

Ligne 270 : Fin du programme

Lignes 280-320 : Sous-programme "attendre"

Résultats du programme:

Si nous entrons 4 valeurs après le lancement du programme, par exemple les nombres 19, 63, 79 et 56, nous obtenons le résultat suivant:

Minimum = 19

Maximum = 79

Moyenne = 54.25

Fin du programme
Ready

Un tel résultat peut bien sûr encore être affiné. Il saute aux yeux en effet que trois valeurs sont supérieures à la moyenne alors qu'une seulement lui est inférieure et que le minimum est bien plus loin de la moyenne que le maximum.

Cela tient à l'inégale répartition des valeurs examinées dans le spectre global de nombres considéré. Rapporté à notre président d'association, cela signifierait que la jeunesse est également représentée dans cette association mais que les personnes plus âgées y dominent nettement.

2.4 L'ordre alphabétique

Restons avec notre président d'association. Supposons maintenant que vous vouliez en tant que président mettre en ordre votre fichier de membres. Vous commencerez certainement par trier les noms dans l'ordre alphabétique. Cela peut être très fastidieux s'il y a une masse importante de membres.

Avec l'aide d'un ordinateur, ce travail ne pose aucun problème. Le programme que nous vous proposons trie un nombre quelconque de mots ou plus généralement de chaînes de caractères. Les données sont ici entrées, contrairement au programme précédent, au moyen de l'instruction DATA. Si vous voulez donc utiliser d'autres données, il vous suffit de modifier les instructions DATA correspondantes et d'entrer le nombre qui convient pour la variable n.

A noter que nous n'avons utilisé que des majuscules pour les données. La conception de l'ordinateur fait en effet que les majuscules seront toujours considérées comme inférieures aux minuscules correspondantes. La condition C < c sera donc par exemple toujours remplie. Si vous voulez donc utiliser des chaînes mixtes avec des majuscules et des minuscules, il vous faut utiliser avant toute comparaison les fonctions LOWER$ ou UPPER$ pour convertir les majuscules en minuscules ou inversement.

Le programme que nous vous présentons peut être aisément modifié pour trier des nombres au lieu de chaînes de caractères. Il vous suffit dans ce cas de transformer les variables w$ et hi$ en w et hi et de préparer les données voulues. Vous pouvez d'ailleurs vous contenter de modifier les données.

Programme:

```
10 REM I3
20 CLS
30 PRINT"I3 - Programme de tri alphabetique"
40 PRINT TAB(6)"de mots"
50 PRINT:PRINT:PRINT
60 PRINT TAB(11)"Bernd Kowal, 1985"
70 GOSUB 260
80 PRINT"Combien de mots le programme"
90 INPUT"doit-il lire ";n
100 IF n=0 THEN 250
110 CLS:DIM w$(n)
120 REM dans ce programme n=7
130 DATA LE,BON,ORDINATEUR,POUR,VOUS
140 DATA LE,CPC 464
150 FOR i=1 TO n
160 READ w$(i):NEXT i
170 FOR i=1 TO n-1
180 FOR j=i+1 TO n
190 IF w$(i)<=w$(j) GOTO 210
200 hi$=w$(j):w$(j)=w$(i):w$(i)=hi$
210 NEXT j
220 PRINT w$(i)
230 NEXT i
240 PRINT w$(n)
250 END
260 REM SP Attendre
270 LOCATE 7,25
280 PRINT"Frappez une touche S.V.P"
290 x$=INKEY$:IF x$="" THEN 290
300 CLS:RETURN
```

Liste de variables:

hi$ = tableau auxiliaire

i = index de comptage

j = index de comptage

n = nombre de variables alphanumériques

w$(i) = mots ou chaînes de caractères

Description du programme:

Lignes 10-70 : Titre

Lignes 80-90 : Entrée du nombre de chaînes

Ligne 100 : Sauter en fin de programme s'il n'y a pas de chaînes à lire

Ligne 110 : L'écran est vidé et de la place est réservée en mémoire

Ligne 120 : Commentaire

Lignes 130-160 : Les données sont préparées et entrées

Lignes 170-240 : Les mots sont triés par ordre alphabétique et le résultat est sorti.
Trier signifie ici que deux mots sont chaque fois comparés entre eux. Le premier mot entré w$(i) avec i=1 est d'abord comparé avec tous les autres mots w$(j) avec j=2...n. Si w$(1) se trouve, dans l'ordre alphabétique, placé après un autre mot w$(j), on échange les deux mots ainsi comparés (ligne 200).

Après ce parcours (lignes 180-210) figure en premier w$(1), le premier mot dans l'ordre alphabétique.
La variable de comptage i est ensuite augmentée chaque fois de 1, c'est-à-dire qu'on compare d'abord le mot figurant en seconde position w$(2) avec tous les autres sauf w$(1). On procède de même pour le troisième mot et ainsi de suite.
C'est ainsi que toutes les comparaisons possibles auront finalement été effectuées.

Ligne 250 : Fin du programme

Lignes 260-300 : Sous-programme "attendre"

Résultats du programme:

FAIT

L'AMSTRAD CPC

L'ORDINATEUR

POUR

VOUS

Une affirmation très juste, triée dans l'ordre alphabétique !

2.5 Jouez-vous au Loto ?

Vous avez 36 ans, vous êtes né le 12.9.48 et vous oubliez votre anniversaire de mariage le 20.3? Vous allez peut-être alors tenter chaque semaine votre chance au Loto avec les nombres suivants:

36, 12, 9, 48, 20, 3

Vous pouvez maintenant oublier ces nombres qui ne vous ont d'ailleurs probablement pas tellement porté chance. Votre CPC va en effet vous livrer chaque semaine 6 nombres différents pour le loto. Malheureusement, la probabilité restant la même, il est douteux que ces nombres "électroniques" vous portent plus chance que votre ancienne combinaison. Pour vous ôter tout espoir, l'ordinateur vous dit d'ailleurs d'emblée combien il y a de possibilités de combinaisons de 6 nombres avec 49 nombres.

Quelques explications mathématiques sont nécessaires pour le programme suivant:

S'il est en effet très aisé de tirer 6 nombres au hasard entre 1 et 49 (avec la fonction RND), la détermination du nombre de possibilités de combinaisons de 6 ou k éléments parmi 49 ou n éléments est plus compliquée puisqu'il nous faut pour cela la grandeur du coefficient binômial n sur k.

C'est la formule suivante qui s'applique:

$$\binom{n}{k} = \frac{n!}{(n-k)! * k!}$$

Pour les lecteurs intéressés par les mathématiques, nous ajouterons que le cofficient binômial correspond aux facteurs fournis par le développement du binôme. Donc:

$$(a+b)^n = a^n + \binom{n}{1} a^{n-1} b + \binom{n}{2} a^{n-2} b^2 + \ldots + \binom{n}{n-1} a b^{n-1} + b^n$$

Mais revenons au problème posé par notre programme. Comme le calcul de probabilités avec les valeurs n=49 et k=6 entraînerait avec la formule ci-dessus des valeurs supérieures à la valeur maximale pouvant être représentée, ce qui provoquerait le message d'erreur "overflow", le quotient de la formule a été réduit dans le programme:

$$\binom{n}{k} = \frac{1 * 2 * 3 * 4 * \ldots * n}{1 * 2 * 3 * \ldots * (n-k) * 1 * 2 * 3 * \ldots * k}$$

$$= \frac{(k+1) * (k+2) * (k+3) * \ldots * n}{1 * 2 * 3 * 4 * \ldots * (n-k)}$$

Si l'on tient compte du fait que $\binom{n}{k} = \binom{n}{n-k}$ comme il est facile de le comprendre d'après la formule de calcul ci-dessus, on obtient pour n=49 et k=6 (ou k=43):

$$\binom{49}{6} = \frac{49 \cdot 44 * 45 * 46 * 47 * 48 * 49}{6 \; 1 * 2 * 3 * 4 * 5 * 6}$$

Ces explications mathématiques devraient en premier lieu vous rendre plus aisée une éventuelle modification du programme. Le programme est conçu de façon à ce qu'une modification de la ligne 190 permette de calculer n'importe quel coefficient binômial. Il faut cependant également modifier la sortie du résultat.

Programme:

```
10 REM I4
20 CLS
30 PRINT"I4 - Programme pour tirer 6 nombres"
40 PRINT TAB(6)"aleatoires entre 1 et 49,"
50 PRINT TAB(6)"et pour determiner le nombre de"
60 PRINT TAB(6)"possibilites de tirer 6"
70 PRINT TAB(6)"nombres parmi 49 (Loto)"
80 PRINT:PRINT:PRINT
90 PRINT TAB(11) "Bernd Kowal, 1985"
100 GOSUB 340
110 PRINT"6 nombres aleatoires entre 1 et 49 :":PRINT
120 FOR i=1 TO 6
130 x(i)=INT(RND*49+1)
140 FOR j=1 TO i-1
150 IF x(i)=x(j) THEN 130
160 NEXT j
170 PRINT "Nombre ";i;":";x(i)
180 NEXT i:PRINT:PRINT
190 n=49:k=6
200 IF k>n THEN z=0:ne=1:GOTO 240
210 IF n-k>k THEN k=n-k
220 aw=n:ew=k:GOSUB 280:z=f
230 aw=n-k:ew=0:GOSUB 280:ne=f
240 PRINT"Il y a";z/ne;"differentes"
250 PRINT"combinaisons possibles de"
260 PRINT"6 nombres parmi 49 !"
270 PRINT:PRINT:PRINT"Fin des calculs":END
280 REM SP Factorielles
290 f=1
300 FOR i=aw TO ew+1 STEP -1
310 f=f*i
320 NEXT i
330 RETURN
340 REM SP Attendre
350 LOCATE 7,25
360 PRINT"Frappez une touche S.V.P"
370 x$=INKEY$:IF x$="" THEN 370
380 CLS:RETURN
```

Liste de variables:

aw = valeur initiale de la boucle

ew = valeur finale de la boucle

f = produit dans le sous-programme "probabilités"

i = index de comptage et facteur dans le sous-programme "probabilités"

j = index de comptage

k = nombre d'éléments à choisir

n = nombre d'éléments

ne = dénominateur dans la formule de calcul de $\binom{n}{k}$

x(i) = nombres aléatoires entre 1 et 49

z = compteur dans la formule de calcul de $\binom{n}{k}$

Description du programme:

Lignes 10-100 : Titre

Lignes 110-180 : 6 nombres aléatoires entre 1 et 49 sont produits puis sortis.
Dans les lignes 140-160, tout nouveau nombre aléatoire produit est comparé à ceux déjà sortis.

En cas d'identité, un nouveau nombre aléatoire x(i) est produit, sans que la variable de comptage soit augmentée.

Ligne 190 : Les variables n et k reçoivent des valeurs

Ligne 200 : Si k>n, $\binom{n}{k}$=0. Dans ce cas, on saute à la sortie du résultat.

Ligne 210 : Si n-k>k, k est fixé égal à n-k pour simplifier le calcul (voir plus haut)

Lignes 220-230 : Calcul des compteur et dénominateur de la formule abrégée

Lignes 240-260 : Sortie du résultat

Ligne 270 : Fin du programme

Lignes 280-330 : Sous-programme "probabilités"
Les probabilités ne sont calculées que si la variable ew égale 0. Sinon des produits quelconques sont formés.

Lignes 340-390 : Sous-programme "attendre"

Résultats du programme:

La sortie des résultats pourrait se présenter ainsi:

6 nombres aléatoires entre 1 et 49:

Nombre 1 : 32
Nombre 2 : 23
Nombre 3 : 5
Nombre 4 : 39
Nombre 5 : 41
Nombre 6 : 7

Il y a 13983816 combinaisons possibles de
6 nombres parmi 49 !

Il ne reste plus qu'à vous souhaiter d'avoir beaucoup de chance !

2.6 Publicité dans une vitrine

Si vous avez un commerce auquel vous soyez prêt à l'occasion de consacrer votre CPC, vous pouvez placer votre ordinateur dans la vitrine du magasin.

Bien entendu, il ne s'agit pas simplement de mettre votre CPC dans la vitrine pour montrer aux gens que vous avez un ordinateur. Il s'agit d'utiliser un programme de publicité. Vous savez en effet que les supports visuels de publicité sont toujours très attractifs.

La présentation d'un programme de publicité dans une vitrine doit être intéressante et attractive puisqu'il faut qu'une personne passant par hasard devant la vitrine soit accrochée au point de s'arrêter pour recevoir les informations publicitaires. Pour obtenir un tel résultat, il est conseillé d'utiliser des couleurs et un texte défilant. Un mélange de graphisme et de lignes de texte peut également être très efficace. On utilisera pour cela les instructions TAG et TAGOFF. On pourrait par exemple créer avec le programme du chapitre 7.2 un graphisme intéressant que l'on intègrerait ensuite dans un programme de publicité. On pourrait même tout simplement intégrer le programme du chapitre 7.2 tout entier comme sous-programme du programme de publicité.

Dans le programme d'exemple du présent chapitre, nous vous présentons cependant une solution plus simple. Après la sortie en grand format des nombres 1, 2 et 3 (rouge sur blanc), le texte publicitaire apparaît trois fois à l'écran, d'abord rouge sur blanc, puis en clignotement bleu-rouge et enfin en clignotement bleu-rouge sur un fond en clignotement blanc-jaune. Les nombres 1, 2 et 3 sont définis conformément aux grilles qui figurent dans le manuel d'utilisation de votre ordinateur. Alors que les grilles du manuel représentent les nombres, comme tous les caractères, par une combinaison de points, nos nombres en grand format sont ici composés de carrés. Ces carrés sont définis avec l'aide de l'instruction WINDOW et remplis avec la couleur fixée pour le fond.

La commande dans le temps du déroulement du programme s'effectue avec les instructions TIME et AFTER. Alors que les nombres en grand format apparaissent chaque fois une demi-seconde à l'écran, le texte de publicité reste chaque fois cinq secondes à l'écran. L'utilisateur peut en outre indiquer combien de temps le programme doit tourner au total. Notez qu'après appel d'un sous-programme avec l'instruction AFTER, le programme est exécuté jusqu'à ce que le temps indiqué soit écoulé.

Le texte publicitaire du programme présenté ici peut bien sûr être modifié à votre guise. Il est également possible de ne pas faire défiler que la dernière ligne. Il pourrait par exemple être très intéressant de faire défiler plusieurs lignes de texte les unes après les autres. En tout cas, cela vaut le coup de faire tourner différentes variantes.

Programme:

```
10 REM I5
20 CLS
30 PRINT "I5 - Programme de publicite en vitrine"
40 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
50 PRINT TAB(9) "Bernd Kowal, 1985"
60 GOSUB 870
70 PRINT"Duree de defilement du spot ?"
80 PRINT:INPUT"Heures   ";st
90 PRINT:INPUT"Minutes  ";mi
100 PRINT:INPUT"Secondes ";se
110 ze=st*3600+mi*60+se+TIME/300
120 INK 2,26:PAPER 2:BORDER 26
130 INK 1,0:INK 0,26:INK 15,1,3:INK 14,26,24
140 CLS
150 FOR i=1 TO 7
160 PAPER #i,3
170 NEXT i
180 WHILE TIME/300<ze
190 GOSUB 270
200 AFTER 25,1 GOSUB 340
210 AFTER 50,2 GOSUB 450
220 AFTER 75,3 GOSUB 560
230 z1=TIME
240 IF TIME<z1+1000 THEN 240
250 WEND
260 MODE 1:END
270 REM Le chiffre 1
280 MODE 1
290 WINDOW #1,12,15,6,8
300 WINDOW #2,16,23,3,20
310 WINDOW #3,8,31,21,23
320 FOR i=1 TO 3:CLS #i:NEXT i
330 RETURN
340 REM Le chiffre 2
350 CLS
360 WINDOW #1,8,15,6,8
370 WINDOW #2,12,27,3,5
380 WINDOW #3,24,31,6,11
390 WINDOW #4,12,27,12,14
400 WINDOW #5,8,15,15,20
410 WINDOW #6,8,31,21,23
420 WINDOW #7,24,31,18,20
430 FOR i=1 TO 7:CLS #i:NEXT i
440 RETURN
450 REM Le chiffre 3
460 CLS
470 WINDOW #1,8,11,6,8
480 WINDOW #2,12,27,3,5
```

```
490 WINDOW #3,24,31,6,11
500 WINDOW #4,16,27,12,14
510 WINDOW #5,24,31,15,20
520 WINDOW #6,12,27,21,23
530 WINDOW #7,8,15,18,20
540 FOR i=1 TO 7:CLS #i:NEXT i
550 RETURN
560 REM Offre speciale
570 MODE 0
580 z2=TIME
590 GOSUB 660
600 IF TIME<z2+1500 THEN 600
610 z3=TIME:PEN 15:CLS:GOSUB 660
620 IF TIME<z3+1500 THEN 620
630 z4=TIME:PAPER 14:BORDER 26,24:CLS:GOSUB 660
640 IF TIME<z4+1500 THEN 640
650 PEN 1:PAPER 2:BORDER 26:RETURN
660 PRINT"  Offre speciale":PRINT:PRINT:PRINT
670 PRINT" Amstrad CPC 464":PRINT:PRINT:PRINT
680 PRINT" avec moniteur":PRINT
690 PRINT" et":PRINT
700 PRINT" lecteur de cassette":PRINT:PRINT:PRINT
710 PRINT" au prix de :":PRINT:PRINT
720 PRINT TAB(7)"4000 FF"
730 a$=" n'est-ce pas fabuleux "
740 z5=TIME
750 IF TIME<z5+300 THEN 750
760 FOR i=1 TO 20
770 b$=RIGHT$(a$,i)
780 LOCATE 1,23
790 PRINT b$;
800 NEXT i
810 FOR i=1 TO 20
820 b$=LEFT$(a$,20-i)
830 LOCATE i,23
840 PRINT b$;
850 NEXT i
860 RETURN
870 REM SP Attendre
880 LOCATE 7,25
890 PRINT"Frappez une touche S.V.P"
900 x$=INKEY$:IF x$="" THEN 900
910 CLS:RETURN
```

Liste de variables:

a$ = chaîne de caractères dans laquelle est placé le texte défilant

b$ = la chaîne de caractères a$ "traitée"

i = variable de comptage

mi = minutes

se = secondes

st = heures

z1
z2
z3 } = durée d'une boucle de temporisation
z4
z5

ze = durée limite en seconde pour le programme tout entier

Description du programme:

Lignes 10-60 : Titre

Lignes 70-100 : Entrée de la durée du déroulement du programme

Ligne 110 : Calcul de la durée limite en seconde pour le déroulement du programme complet. La fonction TIME indique le temps qui s'est écoulé depuis la mise sous tension de la machine (unité: 1/300 de seconde).

Ligne 120 : Les couleurs du fond et du bord sont fixées sur blanc brillant.

Ligne 130 : Les numéros de PEN ou PAPER sont affectés à des couleurs

Ligne 140 : L'écran est vidé

Lignes 150-170 : Fixation de la couleur de fond pour les zones de l'écran restant encore à définir (rouge clair)

Lignes 180-250 : Appel des sous-programmes
Tant que la durée limite n'est pas atteinte, les instructions des lignes 190-240 sont exécutées.

190-240 : Les différents sous-programmes sont appelés en fonction des durées indiquées. En fonction des boucles d'attente du sous-programme Offre spéciale, on peut renoncer en ligne 190 à l'instruction AFTER.

230-240 : Boucle d'attente qui interdit que les lignes 190-220 ne soient traitées avant que les sous-programmes aient été appelés

Ligne 260 : Modification du mode écran (40 caractères par ligne) puisque le sous-programme Offre spéciale travaille avec 20 caractères par ligne. L'écran est en même temps vidé et on atteint la fin du programme.

Lignes 270-330 : Sous-programme pour la sortie en grand format du nombre 1

270-280 : Commentaire et modification du mode écran
290-310 : Définition de fenêtres d'écran carrées qui forment ensemble le nombre 1
320 : Les fenêtres d'écran définies sont remplies de rouge clair (voir lignes 150-170)
330 : Retour du sous-programme

Lignes 340-550 : Sous-programmes pour représenter en grand format les nombres 2 et 3. La procédure est la même que pour le sous-programme précédent.

Lignes 560-860 : Sous-programme "Offre spéciale" pour sortir le texte publicitaire
560-570 : Commentaire et passage en mode 80 colonnes
580-650 : En fonction des variables de durée z2, z3 et z4, un saut est effectué à trois reprises au sous-programme en lignes 660-860 qui sort le texte. Lors des deuxième et troisième sauts en lignes 610 et 630, la couleur du fond ou la couleur du crayon et celle du fond (y compris le cadre) sont modifiées. En ligne 650, on retourne au programme principal.
660-860 : Sous-programme de sortie du texte publicitaire. Après la sortie du texte principal (lignes 660-720) et la définition de la variable a$ (ligne 730) suit une boucle d'attente (lignes 740-750). Le contenu de la chaîne de caractères est ensuite traité en lignes 760-850 puis sorti en défilement.

Lignes 870-920 : Sous-programme "attendre"

2.7 Le puits à idées

- Création de tableaux et de statistiques

- Représentation de courbes et de diagrammes (camemberts, histogrammes, etc...)

- Dessin d'une fonction quelconque

- Biorythmes, analyse de partenaires biorythmiques

- Astrologie et horoscope

- Tests de personnalités ou tests de QI

- Evaluation des réponses à des formulaires

- Gestion d'adresses, gestion de cartes de membre

- Fichier de vocabulaire

- Communications de tout genre

- Banques de données pour n'importe quelles applications

- Gestion scolaire: choix des cours, listes de cours

- Création d'un calendrier

- Cours de dactylographie

- Synthèse de la parole

- Représentation de schémas de fonction

- et bien d'autres choses encore


SUPER
BANK
U-AT 85

3. Voiture et ordinateur

3.1 Remarque préalable

Appartenez-vous à la très nombreuse légion des automobilistes? Si ce n'est pas le cas, nous vous conseillons malgré tout de ne pas négliger ce chapitre. Les programmes qu'il contient sont en effet vraiment intéressants. Ils vous présentent par exemple une utilisation interactive du lecteur de disquette ou de cassette et le programme d'optimisation de l'itinéraire peut vous servir également pour une expédition en vélo.

La notion "interactif" mérite une rapide explication. Elle signifie qu'on accède à partir du programme à la cassette ou à la disquette pour la lire ou pour y écrire. Vous pouvez ainsi sauvegarder durablement des informations que vous pourrez à tout moment examiner ou modifier. Dans la suite de l'ouvrage, vous trouverez encore quelques programmes qui utilisent de cette façon le lecteur de cassette ou de disquette.

3.2 Contrôle sur une longue période de la consommation d'essence

Dans le programme présenté dans ce chapitre les informations à entrer sont la consommation d'essence de votre voiture et la date à laquelle celle-ci a été relevée. Bien entendu vous n'avez pas à rapporter vous-même cette consommation à 100 km. Il suffit bien sûr d'indiquer le nombre de kilomètres parcourus et la quantité absolue de carburant consommée. La conversion en consommation en litres pour 100 km est effectuée dans le programme. Si vous ne vous souvenez plus comment on fait un tel calcul, en voici la formule:

$$\text{Consommation aux 100 km} = \frac{\text{consommation absolue * 100}}{\text{kilomètres parcourus}}$$

Le résultat de ce calcul (il est bien sûr également possible d'effectuer plusieurs calculs consécutifs) peut alors être sauvegardé, avec la date correspondante, sur disquette ou sur cassette. Vous pouvez bien sûr également lire des informations sauvegardées et compléter cette série d'informations (plus précisément: ce fichier). Dans une étape suivante il est également possible de sortir toutes les informations pour pouvoir par exemple déceler une éventuelle "tendance de consommation". La consommation moyenne est également calculée et sortie à cette occasion.

Un mot sur le déroulement du programme:
Les possibilités de sélection pour l'utilisateur du programme sont réalisées en général à travers ce qu'on appelle un menu. Dans ce programme la possibilité de sélection s'exerce différemment, par une suite de questions auxquelles il faut répondre par "o" (pour oui) ou "n" (pour non). Il faut noter à cet égard qu'une fois que vous avez entré un "n", il n'est pas possible de l'annuler (sauf en interrompant le programme en appuyant deux fois sur la touche ESC puis en utilisant l'instruction GOTO pour ne pas perdre les données). Chaque fois en effet que vous lancez le programme (avec RUN) toutes les données entrées précédemment sont perdues.

Cette technique d'interrogation semble au premier abord moins pratique que la technique du menu. Elle présente cependant l'avantage de permettre moins facilement à l'utilisateur d'oublier des étapes importantes du programme - comme par exemple la sauvegarde des données - puisqu'il doit répondre à chaque question de façon active "o" ou "n" avant de poursuivre l'exécution du programme.

Programme:

```
10 REM I6
20 CLS
30 PRINT"I6 - Programme de controle"
40 PRINT TAB(6)"de votre consomation d'essence"
50 PRINT:PRINT:PRINT
60 PRINT TAB(11)"Bernd Kowal, 1985"
70 GOSUB 640
80 DIM v(1000),ta(1000),mo(1000),ja(1000)
90 PRINT"Faut-il charger des donnees"
100 INPUT"sauvegardees (o/n) ";a$:PRINT
110 IF a$="n" THEN 180
120 IF a$<>"o" THEN 90
130 OPENIN "Consomm.":i=1
140 INPUT #9,v(i),ta(i),mo(i),ja(i)
150 IF EOF=-1 THEN 170
160 i=i+1:GOTO 140
170 CLOSEIN
180 CLS:PRINT"Faut-il entrer de nouvelles"
190 INPUT"valeurs (o/n) ";a$:PRINT:PRINT
200 IF a$="n" THEN 370
210 IF a$<>"o" THEN 180
220 i=i+1
230 INPUT"Kilometres parcourus   ";gkm:PRINT
240 INPUT"Consommation en litres ";bv:PRINT
250 PRINT:PRINT"Date  :":PRINT
260 INPUT"Jour    ";ta(i)
270 INPUT"Mois    ";mo(i)
280 INPUT"Annee   ";ja(i):PRINT
290 PRINT:v(i)=bv*100/gkm
300 PRINT"La consommation aux 100 kilometres":PRINT
310 PRINT"est de :",USING "##.##";v(i);:PRINT" litres":P
RINT
```

```
320 PRINT:PRINT"Faut-il entrer d'autres valeurs"
330 INPUT"(o/n) ";a$
340 IF a$="n" THEN 370
350 IF a$<>"o" THEN 320
360 CLS:GOTO 220
370 CLS: PRINT"Faut-il stocker toutes les stats "
380 PRINT"de consommation (y compris les valeurs"
390 PRINT"eventuellement chargees auparavant)"
400 INPUT"avec la date (o/n) ";a$
410 IF a$="n" THEN 480
420 IF a$<>"o" THEN 370
430 OPENOUT "Consomm."
440 FOR j=1 TO i
450 PRINT #9,v(j),ta(j),mo(j),ja(j)
460 NEXT j
470 CLOSEOUT
480 CLS:PRINT"Voulez-vous examiner toutes les"
490 INPUT"stats en une fois (o/n) ";a$
500 IF a$="n" THEN 630
510 IF a$<>"o" THEN 480
520 CLS:PRINT" Consommation",TAB(28)"Date":PRINT
530 FOR j=1 TO i
540 PRINT;USING "##.##";v(j);:PRINT" litres ",
550 PRINT ;USING "##.";ta(j),
560 PRINT ;USING "##.";mo(j),
570 PRINT ;USING "##";ja(j)
580 IF j=15 THEN GOSUB 640
590 sv=sv+v(j):NEXT j
600 PRINT:PRINT"Soit une consommation moyenne de :"
610 PRINT USING "##.##";sv/i;
620 PRINT" litres"
630 PRINT:PRINT:PRINT:PRINT"Fin du Programme":END
640 REM SP Attendre
650 LOCATE 7,25
660 PRINT"Frappez une touche S.V.P"
670 x$=INKEY$:IF x$="" THEN 670
680 CLS:RETURN
```

Liste de variables:

a$ = chaîne de réponse (o/n)

bv = consommation d'essence absolue

gkm = kilomètres parcourus

i = variable de comptage

j = variable de comptage

ja(i) = année

mo(i) = mois

sv = tableau d'addition pour v(i)

ta(i) = jour

v(i) = consommation aux 100 km

Description du programme:

Lignes 10-70 : Titre

Ligne 80 : On réserve de la place en mémoire pour les variables indexées

Lignes 90-120 : Première question "Lire"
Cette question, de même que les suivantes, est sortie à nouveau si l'utilisateur n'entre ni "o" ni "n". Si "n" est entré, on saute à la question suivante (ligne 180).

Ligne 130 : Le fichier d'entrée "Consomm." est ouvert et la variable de comptage i est fixée à 1.

Ligne 140 : Lecture des informations stockées

Ligne 150 : Si la fin du fichier n'a pas été atteinte (EOF = -1), on saute à la ligne 170

Ligne 160 : La variable de comptage i est augmentée de 1 et on retourne en ligne 140 pour la lecture d'autres données

Ligne 170 : Fermeture du fichier d'entrée

Lignes 180-210 : Seconde question "Entrer"

Ligne 220 : La variable de comptage i est augmentée

Lignes 230-280 : On demande les kilomètres parcourus, la consommation d'essence absolue et les indications concernant la date

Lignes 290-310 : Calcul de la consommation aux 100 km et sortie du résultat avec une précision de deux décimales après la virgule

Lignes 320-360 : Interrogation sur une entrée supplémentaire, vidage de l'écran et le cas échéant saut en ligne 220

Lignes 370-420 : Troisième question "Sauvegarder" (les lignes 320-360 appartiennent pour ainsi dire à la seconde question)

Lignes 430-470 : Le fichier de sortie "Consomm." est ouvert, des informations y sont sauvegardées (il y a chaque fois i valeurs différentes dans la mémoire de travail de l'ordinateur) et le fichier de sortie est refermé

Lignes 480-510 : Quatrième question "Resultats"

Ligne 520 : Sortie du titre du tableau

Lignes 530-590 : Les valeurs de consommation sont sorties formatées avec la date correspondante. Après 15 sorties, un saut est effectué au sous-programme "attendre" pour que les résultats ne disparaissent pas vers le haut de l'écran. Les différentes valeurs de consommation sont additionnées dans le tableau sv (ligne 590) pour pouvoir plus tard calculer la moyenne arithmétique de toutes les valeurs de consommation.

Lignes 600-620 : Calcul de la moyenne arithmétique et sortie du résultat

Ligne 630 : Fin du programme

Lignes 640-680 : Sous-programme "attendre"

Résultats du programme:

Après le lancement du programme, des valeurs sont lues et/ou entrées (et éventuellement sauvegardées directement). La sortie du résultat peut ensuite se présenter à l'écran de la façon suivante:

```
Consommation Date

8.05 litres  1.2.85
8.55 litres  9.2.85
8.13 litres  20.2.85
7.81 litres  1.3.85

Soit une consommation
moyenne de : 8.14 litres
```

3.3 Optimisation de l'itinéraire

La voiture sera à nouveau au centre de nos préoccupations. Il s'agit toujours de consommation de carburant, mais cette fois de façon indirecte.

L'objet de ce chapitre est de planifier un voyage. Il faut indiquer les différentes localités que vous voulez parcourir et les distances entre ces localités. Votre CPC vous dira alors dans quel ordre vous devrez parcourir ces localités pour faire le moins de kilomètres possible. Il s'agit donc d'économiser l'énergie de votre voiture ainsi que la vôtre.

Il n'est toutefois pas si aisé de trouver un algorithme (procédure se répétant d'après un schéma déterminé) approprié à la solution de ce problème. Vous pensez peut-être que votre ordinateur devrait prendre en compte tous les ordres possibles. Et c'est effectivement ce qu'il faudrait faire dans le cas où seul un petit nombre de localités serait entré. Il y a par exemple pour une entrée de trois localités a, b et c (le lieu de départ reste le même) deux possibilités d'ordre différentes: a-b-c et a-c-b. Si vous entrez 4 localités a, b, c et d, il y a 6 possibilités différentes: a-b-c-d, a-b-d-c, a-d-c-b, a-d-b-c, a-c-b-d et a-c-d-b.

Ces différents ordres ne se distinguent que par l'emplacement des éléments (ou localités). Le nombre d'ordres de n éléments, c'est-à-dire le nombre de possibilités différentes pour parcourir les n localités indiquées égale n! (n! = 1 * 2 * 3 * ... * n). Si vous calculez par exemple 10!, vous obtenez: 10! = 3628800. S'il fallait donc envisager réellement toutes les possibilités, cela conduirait certainement à des temps de calcul beaucoup trop longs.

On obtient en général un résultat raisonnable en allant consécutivement d'une localité à la plus proche. Il se peut cependant que 4 localités soient ainsi réparties dans l'espace:

A

D

B

C

En partant de A on obtiendrait l'ordre A - B - C - D alors que c'est l'ordre A - C - B - D qui est le plus avantageux.

C'est pourquoi le programme suivant tient compte de la possibilité que, lors du classement de chaque nouvelle localité parmi les localités déjà ordonnées, il soit plus avantageux de ne pas aller tout de suite dans la localité la plus proche mais d'abord dans la deuxième plus proche.

Le programme compare ainsi entre eux n-1 (n = nombre de localités entrées, y compris le point de départ) itinéraires et sort en résultat le choix le plus avantageux.

Appliqué à l'exemple donné plus haut, les ordres suivants seraient examinés:

A - B - C - D
A - C - B - D
A - B - D - C

La localité figurant en seconde et en troisième position est donc toujours au moins une fois non pas la localité la plus proche mais la plus proche en second.

Cet algorithme trouvera en général l'itinéraire le plus avantageux pour votre voyage et vous libèrera ainsi d'un travail qui peut être fastidieux si vous voulez parcourir de nombreuses localités au cours de votre voyage.

Programme:

```
10 REM I7
20 CLS
30 PRINT"I7 - Programme d'optimisation"
40 PRINT TAB(6)"d'itineraire"
50 PRINT:PRINT:PRINT
60 PRINT TAB(11) "Bernd Kowal, 1985"
70 GOSUB 520
80 PRINT"Nombre total de villes a prendre"
90 INPUT"en compte ";n:PRINT
100 DIM h$(n),s$(n),o$(n),e(n,n),fs(n)
110 PRINT"Veuillez entrer les noms de ville"
120 PRINT"ou un symbole de ville :":PRINT
130 INPUT"D'abord la ville de depart ";o$(1):PRINT
140 FOR i=2 TO n
150 PRINT"Nom de ville No ";i;:INPUT o$(i)
160 NEXT i:CLS
170 PRINT"Veuillez entrer les distances suivantes"
180 PRINT"en kilometres :":PRINT
190 FOR i=1 TO n-1
200 FOR j=i+1 TO n
210 PRINT o$(i);"-";o$(j),:INPUT e(i,j)
220 e(j,i)=e(i,j)
230 NEXT j:NEXT i:CLS
240 FOR u=1 TO n-1
250 FOR i=2 TO n:s$(i)="":NEXT i
260 i=1:s$(i)=o$(i):l=2
270 kw=10^30
280 FOR j=1 TO n
290 FOR k=1 TO n
300 IF s$(k)=o$(j) THEN 330
310 NEXT k
320 IF e(i,j)<kw THEN kw=e(i,j):q=j
330 NEXT j
340 IF u=l AND kr=0 THEN e(i,q)=10^30:kr=1:GOTO 270
350 s$(l)=o$(q)
360 fs(u)=fs(u)+kw
370 IF l<n THEN i=q:l=l+1:GOTO 270
380 IF u>1 THEN 410
390 FOR i=1 TO n:h$(i)=s$(i):NEXT i
400 fs=fs(u):GOTO 440
410 IF fs(u)>=fs THEN 440
420 FOR i=1 TO n:h$(i)=s$(i):NEXT i
430 fs=fs(u)
440 kr=0:NEXT u
450 PRINT"L'itineraire optimal est le suivant :":PRINT
460 FOR i=1 TO n
470 IF i=n THEN PRINT h$(i):GOTO 490
480 PRINT h$(i);"-";
```

```
490 NEXT i
500 PRINT:PRINT"Soit";fs;"kilometres."
510 PRINT:PRINT:PRINT"Fin des calculs":END
520 REM SP Attendre
530 LOCATE 7,25
540 PRINT"Frappez une touche S.V.P"
550 x$=INKEY$:IF x$="" THEN 550
560 CLS:RETURN
```

Liste de variables:

e(i,j) = distance entre deux localités

fs = nombre de kilomètres pour l'itinéraire optimal

fs(i) = nombre de kilomètres pour une possibilité d'itinéraire

h$(i) = les localités dans l'ordre optimal

i = variable de comptage

j = variable de comptage

k = variable de comptage

kr = variable de contrôle pour le rangement de la localité la plus proche en second

kw = distance la plus courte d'une localité à une localité qui n'a pas encore été ordonnée

l = variable de comptage pour les n-1 ordonnancements (pour chaque possibilité d'itinéraire)

n = nombre de localités

o$(i) = les noms de localités entrés

q = index pour le rangement de la localité la plus proche ou de la deuxième plus proche

s$(i) = les localités ordonnées pour chaque possibilité d'itinéraire

u = variable de comptage pour les n-1 possibilités d'itinéraire

Description du programme:

Lignes 10-70 : Titre

Lignes 80-90 : Entrée du nombre de localités

Ligne 100 : On réserve de la place en mémoire pour les variables indexées

Lignes 110-160 : Entrée des noms de localités

Lignes 170-230 : Entrée des distances
En ligne 220, une distance e(i,j) est fixée égale à la distance non entrée e(j,i) car dans la boucle en lignes 280-330 la variable de comptage j varie de 1 à n, indépendamment de la grandeur de la variable i.

Ligne 240 : Ouverture de la boucle de programme pour les n-1 possibilités d'itinéraire

Ligne 250 : La variable s$(i) est "libérée" des anciennes affectations pour être disponible pour la nouvelle possibilité d'itinéraire

Lignes 260-270 : Définition des variables i, l, s$(1) et kw (s$(1) = localité de départ o$(1). On suppose pour la variable kw la valeur arbitraire très élevée 10^30 qui ne doit pas être atteinte lors des indications de distance.

Lignes 280-330 : La plus petite distance d'une localité à une localité "non encore parcourue" j est recherchée et stockée provisoirement dans le tableau kw. On enregistre dans le tableau q quelle localité a été affectée à j. On recherche en lignes 290-310 si la localité actuelle a déjà été affectée ou "parcourue".

Ligne 340 : Lors du premier parcours, on cherche chaque fois quelles sont les distances les plus courtes. Les parcours suivants se distinguent par le fait que les distances les plus courtes e(i,q) sont fixées égales à 10^30 en des endroits toujours différents (en fonction de u et l) pour pouvoir rechercher à nouveau la plus courte distance (c'est alors bien sûr la deuxième plus courte puisque la distance la plus courte a reçu la valeur 10^30). La variable de contrôle kr permet que cette procédure ne soit exécutée qu'une seule fois.

Lignes 350-360 : La localité trouvée est stockée provisoirement dans le tableau s$(i) et la distance en kilomètres trouvée est additionnée à la valeur du tableau fs(u).

Ligne 370 : Si l<n, la procédure de rangement est poursuivie pour chaque itinéraire possible

Lignes 380-400 : Lors du premier parcours (alors u = 1) l'itinéraire trouvé est considéré comme optimal (définition des variables h$(i) et fs) pour des comparaisons ultérieures (ligne 410)

Lignes 410-430 : Le nombre de kilomètres de la possibilité d'itinéraire trouvée est comparé à la valeur correspondante du parcours précédent. Le cas échéant, les variables h$(i) et fs reçoivent les nouvelles valeurs.

Ligne 440 : La variable de contrôle kr est fixée à 0 et un nouveau parcours est effectué (à partir de la ligne 240)

Lignes 450-500 : Sortie du résultat

Ligne 510 : Fin du programme

Lignes 520-570 : Sous-programme "attendre"

Résultats du programme:

Entrons 6 localités réparties géographiquement de la façon suivante (D est le point de départ):

B

D C

F

A

E

avec les distances exactes suivantes:

	A	B	C	D	E	F
A		40	46	13	39	12
B			30	22	49	32
C				30	26	29
D					34	12
E						23
F						

Nous obtiendrons:

Voici l'itinéraire optimal:

D - A - F - E - C - B

Ce qui représente 104 kilomètres.

Par rapport aux programmes précédents, l'augmentation du délai de calcul est évidente. Si vous utilisez ce programme pour un nombre relativement important de localités, vous devez tout de même faire preuve d'un peu de patience.

3.4 Les coûts de la voiture

Vous trouverez au chapitre 6.2 un programme volontairement très général de budget. Comme il vous permet de créer vous-même des catégories qui peuvent ensuite être sorties individuellement, ce programme peut parfaitement être également utilisé pour le domaine des coûts de la voiture. Les catégories possibles sont par exemple: assurance, impôts, dépenses d'essence, réparations, entretien, amendes, etc. Si vous voulez utiliser ce programme exclusivement pour les coûts de votre voiture, une extension du programme pour calculer le coût moyen au kilomètre serait très intéressante. Il y a plusieurs possibilités d'intégrer les indications de kilométrage dans ce programme:

1) Entrée des kilomètres au besoin, par une section particulière du programme pouvant être appelée à partir du menu

2) Entrée du kilométrage actuel lors de chaque dépense d'argent

3) Lors de l'entrée d'une période pour la section "calculs" du programme, une valeur pour le kilométrage est réclamée et le coût moyen au kilomètre est sorti immédiatement

La première possibilité présente l'inconvénient que la section de programme "calculs" n'utilisera pas toujours le nombre exact de kilomètres parcourus puisque le kilométrage n'est pas actualisé lors de chaque entrée. La seconde possibilité élimine cet inconvénient mais il est à l'inverse improbable que vous ayez toujours en tête le kilométrage actuel de votre voiture. La troisième possibilité est la plus simple à intégrer du point de vue de la technique de programmation et elle élimine en outre les inconvénients évoqués. Vous n'avez plus en effet à noter le kilométrage que de temps en temps. D'ailleurs pour les coûts de la voiture, il est de toute façon judicieux d'indiquer dans la section "calculs" du programme des périodes de l'ordre de l'année ou de la demi-année puisque les impôts ou les primes d'assurance ne tombent également qu'une fois ou deux par an.

Pour modifier ainsi le programme de budget, vous devez ajouter les lignes suivantes:

```
1395 PRINT:PRINT:PRINT"Combien de kilometres avez-vous"
1400 INPUT"parcouru durant cette periode ";km
1514 PRINT:PRINT:PRINT"Rapporte au kilometre,"
1515 PRINT"cela donne un cout de"
1516 PRINT INT(su/km*100+0.5)/100;"FF par Km."
1517 GOSUB 2060
```

Seule la ligne 1400 en efface une autre. On peut cependant facilement renoncer au commentaire "Je calcule".

Nous vous souhaitons donc une utilisation profitable du programme et ... de ne pas tomber à la renverse quand vous découvrirez ce que vous coûte votre voiture.

3.5 Le puits à idées

- Simulation du parcours d'une Porsche 935 de 700 chevaux sur le nouveau Nürburgring.

- Enregistrement précis d'une course (le passage d'un véhicule peut être détecté avec des cellules photo-électriques).

- Autos réalisées avec l'ordinateur (dessin, objets vidéo commandés électroniquement, tests d'endurance avec un simulateur).

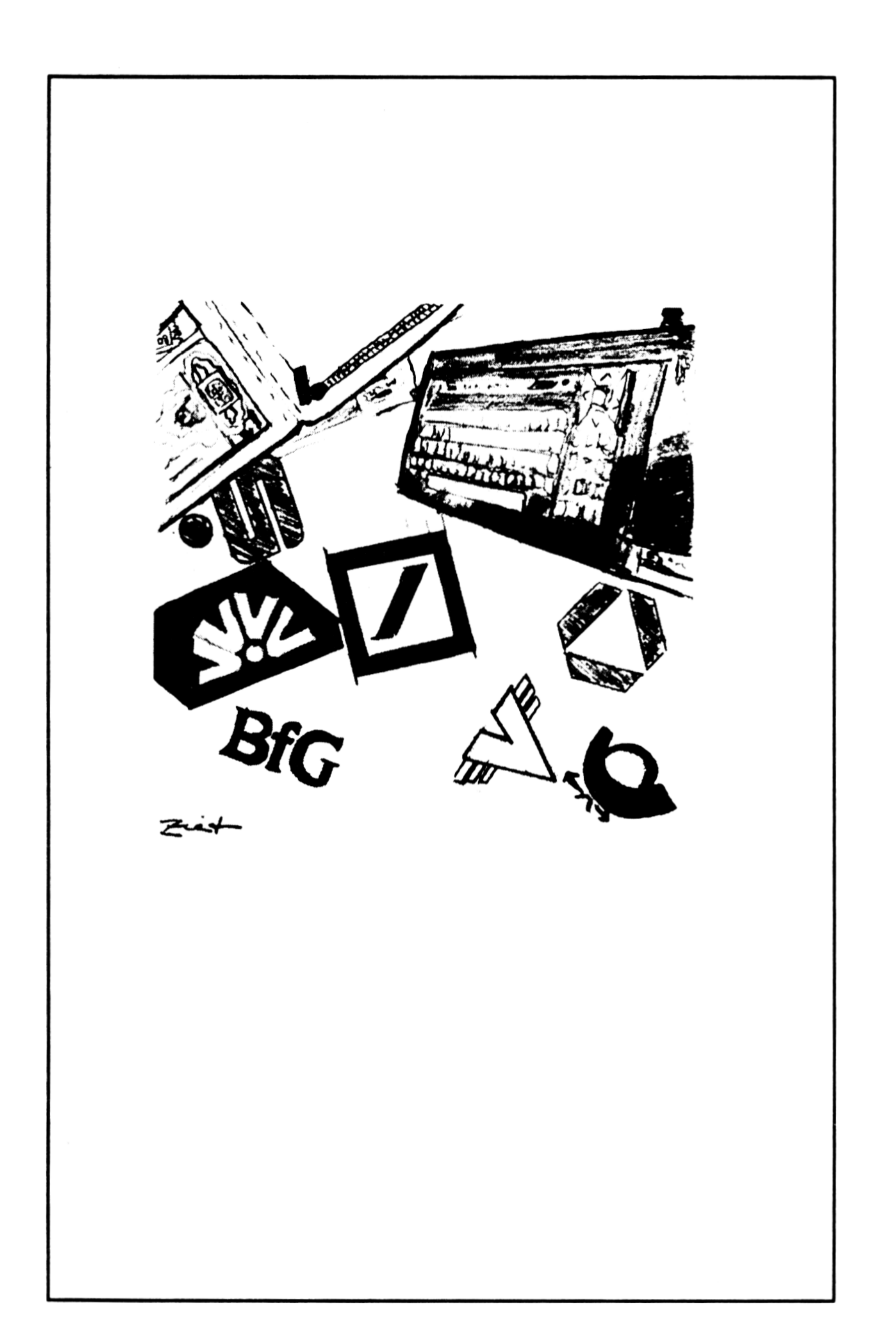
- Entraînement à la circulation (maîtrise de situations de circulation simulées).

- Ecole de conduite: élèves, cours, comptes et résultat journalier.

- Banque de données pour des comparaisons entre automobiles (vitesse maximum, accélération, résultats de tests etc.).

- Tests de connaissances géographiques avec l'ordinateur.

- Et bien d'autres choses encore.


BfG

4. Argent et crédit

4.1 Remarque préalable

Lorsqu'il s'agit d'argent, cela devient sérieux et la plupart des gens veulent tout savoir de façon très précise.

Les caisses d'épargne et les banques utilisent depuis longtemps des ordinateurs pour calculer vite et sans erreur les problèmes les plus compliqués. Vous devez en tant que client vous contenter le plus souvent des indications faites par l'institut de crédit, par exemple lorsqu'il s'agit d'obtenir un crédit. Ces indications peuvent parfois être confuses ou très différentes d'un institut à l'autre de sorte qu'une comparaison des offres risque d'être très difficile. C'est pourquoi nous allons essayer dans ce chapitre de mieux armer le client des banques de façon à arriver à une plus grande égalité des chances.

Naturellement, les problèmes et programmes que nous vous présentons ici ne peuvent se comparer aux logiciels très puissants des grands instituts de crédit. Ils peuvent cependant vous être utiles si vos entrées et vos sorties d'argent ne sont pas toujours équilibrées et si vous êtes parfois amené soit à demander un crédit soit à placer de l'argent.

4.2 Le calcul des intérêts

La base de tout ce qui touche à l'argent et le crédit est le calcul des intérêts. D'un point de vue purement mathématique, c'est là un problème assez simple. Peut-être certains lecteurs ne se souviennent-ils cependant pas comment les grandeurs élémentaires du calcul d'intérêt sont calculées. C'est pourquoi le programme suivant vise plus à vous permettre de vous exercer au maniement de ces grandeurs qu'à une application pratique pour vos plans de financement.

Les formules de calcul suivantes sont à la base du programme:

$$\text{Intérêt} = \frac{\text{Capital} * \text{Temps} * \text{Taux d'intérêt}}{100}$$

$$\text{Capital} = \frac{\text{Intérêt} * 100}{\text{Taux d'intérêt} * \text{Temps}}$$

$$\text{Taux d'intérêt} = \frac{\text{Intérêt} * 100}{\text{Capital} * \text{Temps}}$$

$$\text{Temps} = \frac{\text{Intérêt} * 100}{\text{Capital} * \text{Taux d'intérêt}}$$

L'indication du temps se fait ici en années. Une année a 360 jours ou 12 mois de 30 jours. Si vous voulez prendre en compte les mois ou/et les jours, vous devez effectuer les calculs de conversion appropriés.

Programme:

```
10 REM I8
20 CLS
30 PRINT"I8 - Programme de calcul d'interets"
40 PRINT TAB(6)"de taux d'interet, du capital ou"
50 PRINT TAB(6)"de la duree d'immobilisation"
60 PRINT TAB(6)"du capital"
70 PRINT:PRINT:PRINT
80 PRINT TAB(11) "Bernd Kowal, 1985"
90 GOSUB 460
100 PRINT TAB(12)"Selection":PRINT:PRINT
110 PRINT"Que faut-il calculer ?":PRINT
120 PRINT TAB(30)"Entree":PRINT
130 PRINT"Interets",TAB(33)"1":PRINT
140 PRINT"Taux d'interet",TAB(33)"2":PRINT
150 PRINT"Capital",TAB(33)"3":PRINT
160 PRINT"Duree d'immobilisation"
170 PRINT"du capital",TAB(33)"4"
180 PRINT:PRINT:PRINT:INPUT"Votre choix";a
190 IF a<1 OR a>4 THEN CLS:GOTO 100
200 CLS:PRINT"Entree des valeurs connues :":PRINT
210 IF a<>1 THEN INPUT"Interets (en FF)       ";Zi
220 IF a<>2 THEN INPUT"Taux d'interet (en pourcentage) "
;Zs
230 IF a<>3 THEN INPUT"Capital (en FF)        ";K
240 IF a<>4 THEN INPUT"Immobilisation (en annees) ";Zd
250 PRINT:PRINT
260 IF a>1 THEN 310
270 Zi=K*Zd*Zs/100
280 PRINT"En";Zd;"annees, le capital produira"
290 PRINT Zi;"FF d'interets."
300 GOTO 410
310 IF a>2 THEN 350
320 Zs=Zi*100/K/Zd
330 PRINT"Le taux d'interet est de";Zs;"%."
340 GOTO 410
350 IF a>3 THEN 390
360 K=Zi*100/Zs/Zd
370 PRINT"Le capital s'eleve a";K;"FF."
380 GOTO 410
390 Zd=Zi*100/K/Zs
400 PRINT"L'immobilisation est de";Zd;"annees."
410 PRINT:PRINT:INPUT"Encore un calcul (o/n) ";a$
420 IF a$="o" THEN CLS:GOTO 100
430 IF a$<>"n" THEN 410
440 PRINT:PRINT:PRINT"Alors j'ai fini !"
450 END
460 REM SP Attendre
470 LOCATE 7,25
```

```
480 PRINT"Frappez une touche S.V.P"
490 x$=INKEY$:IF x$="" THEN 490
500 CLS:RETURN
```

Liste de variables:

a = variable de réponse pour le menu

a$ = chaîne de réponse (o/n)

K = capital

Zd = durée de l'immobilisation du capital

Zi = intérêt

Zs = taux d'intérêt

Description du programme:

Lignes 10-90 : Titre

Lignes 100-190 : Sélection dans le menu

Lignes 200-250 : Des valeurs sont demandées pour les grandeurs connues

Lignes 260-400 : Les grandeurs inconnues sont calculées et le résultat est sorti

Lignes 410-440 : Saut au menu s'il reste encore un calcul à exécuter. Sinon sortie d'un commentaire.

Ligne 450 : Fin du programme

Lignes 460-510 : Sous-programme "attendre"

Résultats du programme:

Si nous entrons dans le menu un 1, on nous demande ensuite les valeurs correspondant au taux d'intérêt, au capital et à la durée. Si nous prenons un taux d'intérêt de 8.95%, un capital de 347 F et une durée de 4 ans, l'ordinateur nous annonce:

Au bout de 4 ans, le capital
produira 124.226 F d'intérêts

Un autre calcul (o/n) ?

4.3 L'argent de votre compte d'épargne (calcul d'intérêts des intérêts)

Si vous portez un certain capital dans une banque, au bout d'un an des intérêts s'ajouteront à ce capital. L'année suivante - et cela n'a pas été pris en compte au chapitre précédent - les intérêts déjà produits produiront eux-mêmes des intérêts (intérêts des intérêts). C'est pourquoi le programme du chapitre précédent se prête mal à une application pratique. Le programme que nous vous proposons ici calcule lui par contre exactement comment votre argent "grandit" sur votre compte d'épargne. Il vous suffit d'indiquer une somme de départ et un taux d'intérêt quelconque. Vous obtenez alors en résultat le montant final après un nombre d'années quelconque.

Programme:

```
10 REM I9
20 CLS
30 PRINT"I9 - Programme de calcul d'interets des"
40 PRINT TAB(6)"interets avec des interets payes"
50 PRINT TAB(6)"annuellement, pour un capital"
60 PRINT TAB(6)"de depart donne"
70 PRINT:PRINT:PRINT
80 PRINT TAB(11) "Bernd Kowal, 1985"
90 GOSUB 260
100 INPUT"Capital de depart          ";Ak
110 INPUT"Taux d'interet annuel      ";Zs
120 PRINT"Duree d'immobilisation"
130 INPUT"du capital en annees       ";Zd
140 j=20:l=1
150 CLS:PRINT" Annee        Interet             Capital"
160 PRINT:FOR i=l TO Zd
170 Zb=Ak*Zs/100
180 Ek=Ak+Zb
190 PRINT; USING "####    ";i;
200 PRINT; USING "#########.##    ";Zb;
210 PRINT; USING "###########.##";Ek
220 Ak=Ek
230 IF i=j THEN l=j+1:j=j+20:GOSUB 260:GOTO 150
240 NEXT i
250 END
260 REM SP Attendre
270 LOCATE 7,25
280 PRINT"Frappez une touche S.V.P"
290 x$=INKEY$:IF x$="" THEN 290
300 CLS:RETURN
```

Liste de variables:

Ak = capital de départ

Ek = capital final

i = variable de comptage

j = variable auxiliaire qui permet une interruption du programme lorsque l'écran est plein

l = valeur variable de début de boucle

Zb = montant des intérêts par an

Zd = durée de l'immobilisation du capital

Zs = taux d'intérêt annuel en pourcentage

Description du programme:

Lignes 10-90 : Titre

Lignes 100-130 : Demande des données de départ

Ligne 140 : La valeur de début de boucle est fixée sur 1 et la variable auxiliaire j sur 20

Ligne 150 : L'écran est vidé et un titre de tableau est sorti

Lignes 160-240 : On calcule pour chaque année le montant des intérêts et le capital final. Les résultats sont sortis formatés. Après la sortie d'une ligne de résultat, le capital final calculé devient le capital de départ pour la période suivante (ligne 220).

La ligne 230 permet l'interruption du programme après 20 périodes (sous-programme "attendre"). Dans ce cas, la valeur de départ de la boucle 1 est modifiée en conséquence.

Ligne 250 : Fin du programme

Lignes 260-310 : Sous-programme "attendre"

Résultats du programme:

Pour un capital de départ de 1000 F, un intérêt annuel de 4.8% et une durée de 10 années, on obtient:

Année	Intérêt	Capital
1	48.00	1048.00
2	50.30	1098.30
3	52.72	1151.02
4	55.25	1206.27
5	57.90	1264.17
6	60.68	1324.85
7	63.59	1388.45
8	66.65	1455.09
9	69.84	1524.94
10	73.20	1598.13

4.4 L'intérêt annuel effectif

Lorsque vous interrogez un institut de crédit sur ses propositions de crédit, les conseillers vous fournissent les offres les plus diverses. Sans indication de l'intérêt annuel effectif, il est difficile voire impossible de comparer les offres. L'intérêt effectif indique en effet quelle offre de crédit est la plus avantageuse.

Bien entendu, les instituts de crédit calculent pour leur comptabilité interne l'intérêt effectif. Mais un certain nombre de banques se font tirer l'oreille lorsqu'il s'agit de communiquer à leurs clients l'intérêt effectif.

Mais qu'est-ce exactement que l'intérêt effectif et comment est-il calculé?

L'intérêt effectif intègre tous les éléments du prix d'un crédit. C'est en quelque sorte une mesure du prix d'un crédit.

Suivant la méthode de calcul utilisée, le calcul de l'intérêt annuel effectif est plus ou moins simple. Avec la méthode "mensuelle", le problème est vraiment simple à résoudre:

$$e = \frac{24 * (\text{taux d'intérêt mensuel} * \text{durée en mois} + b)}{\text{durée en mois} + 1}$$

où:

e = taux d'intérêt effectif en % par an

b = frais de gestion en % du montant originel du crédit

Tout se complique si la méthode utilisée est la méthode des 360 jours. C'est l'équation suivante qui permet de calculer l'intérêt annuel effectif pour un crédit avec paiement mensuel d'intérêts et calcul mensuel des intérêts (par rapport au montant du crédit):

$$\frac{1 + \frac{b}{100}}{M} + \frac{p}{100} = \frac{q^{j}}{(\frac{12}{i}\ \frac{11}{2}) * (q^{j} - 1) + (1+\frac{m-1}{24}*i) * \frac{m}{1+\frac{m}{12} *1}}$$

où:

e = taux d'intérêt effectif recherché en % par an (100 * i)

i = e/100

q = 1+i

M = durée en mois
= nombre de taux mensuels à réunir (= 12*J+m)

J = durée en années pleines
= nombre de fois qu'apparaît M/12

m = durée restante en mois = M - 12 * J

p = taux d'intérêt nominal par mois en % du montant original du crédit

b = frais de gestion en % du montant originel du crédit

Cette équation ne peut être réduite à e = ... Si l'on excepte des solutions mathématiques d'approche très compliquées, e ne peut être déterminé que par des tâtonnements successifs. Ce n'est pas pour rien que des ordinateurs sont utilisés dans les banques pour déterminer l'intérêt effectif.

Si la formule du calcul mensuel est accessible même aux non-mathématiciens, la méthode des 360 jours a l'avantage de fournir des résultats plus précis.

Avec l'aide de votre CPC et un programme approprié, la détermination du taux d'intérêt effectif n'est pas si difficile. Nous partons pour cela du principe suivant:
Si nous amenons les termes à gauche du signe égale à droite de ce signe (en effectuant les changements de signe qui s'imposent), nous obtenons une équation de la forme

0 = ...

Si nous essayons maintenant différentes valeurs pour e, alors l'expression à droite du signe égale est inférieure à 0 si la valeur choisie pour e est trop petite. Elle est par contre supérieure à 0 si nous choisissons une valeur trop grande.

Le programme suivant choisit donc des valeurs toujours plus grandes pour e jusqu'à ce que l'expression à la droite du signe égale soit supérieure à 0. Pour approcher de plus près la valeur exacte de e on choisit pour les valeurs-tests des intervalles et des écarts de plus en plus réduits de façon à obtenir au bout du compte un résultat précis à 4 décimales près après la virgule.

Une telle méthode n'est pratiquement possible qu'avec un ordinateur. Sinon le temps perdu dans les tâtonnements serait beaucoup trop important. Vous devez indiquer au programme simplement les valeurs pour les grandeurs M, p et b de l'équation donnée plus haut.

Programme:

```
10 REM I10
20 CLS
30 PRINT"I10 - Programme de calcul du taux"
40 PRINT TAB(7)"d'interet effectif"
50 PRINT:PRINT:PRINT
60 PRINT TAB(11) "Bernd Kowal, 1985"
70 GOSUB 310
80 PRINT"Nombre de mensualites"
90 INPUT"convenues ";m:PRINT
100 PRINT"Taux d'interet mensuel nominal en % du"
110 INPUT"montant originel du credit ";p:PRINT
120 PRINT"Frais de gestion en % du"
130 INPUT"montant originel du credit  ";b
140 IF m=0 OR p=0 THEN CLS:GOTO 80
150 PRINT:PRINT:PRINT:PRINT
160 k1=1:k2=100:s=1
170 FOR e=k1 TO k2 STEP s
180 i=e/100:q=1+i:j=FIX(m/12):rm=m-12*j
190 lt=(1+b/100)/m+p/100:z=q^j
200 n1=(12/i+11/2)*(q^j-1)
210 n2=(1+(rm-1)/24*i)*rm/(1+rm/12*i)
220 vg=z/(n1+n2)-lt
230 IF vg>0 THEN GOSUB 280:GOTO 170
240 NEXT e
250 PRINT"Taux d'interet annuel effectif :":PRINT
260 PRINT"e = ";USING "##.####";e;:PRINT"%"
270 PRINT:PRINT:PRINT"Fin des calculs":END
280 k1=e-s:k2=e:s=s/10
290 IF s=0.000001 THEN 250
300 RETURN
310 REM SP Attendre
320 LOCATE 7,25
330 PRINT"Frappez une touche S.V.P"
340 x$=INKEY$:IF x$="" THEN 340
350 CLS:RETURN
```

Liste de variables:

b = frais de gestion

e = taux d'intérêt effectif recherché

i = e/100

j = durée en années pleines

k1 = valeur initiale de la boucle du programme (limite inférieure de l'intervalle)

k2 = valeur finale de la boucle du programme (limite supérieure de l'intervalle)

lt = partie gauche de l'équation (à gauche du signe égale)

m = durée en mois

n1 = premier opérande du dénominateur (à droite du signe égale)

n2 = second opérande du dénominateur (à droite du signe égale)

p = taux d'intérêt nominal par mois

q = 1+i

rm = durée restante en mois

s = pas de la boucle du programme

vg = valeur à comparer (supérieure ou inférieure à 0)

z = compteur (à droite du signe égale)

Description du programme:

Lignes 10-70 : Titre

Lignes 80-130 : On demande les valeurs de départ

Ligne 140 : On demande à nouveau une valeur si un 0 a été entré pour m ou pour p

Ligne 150 : Sortir quatre lignes vides

Ligne 160 : On affecte une valeur de départ aux valeurs initiale, finale et au pas de la boucle de programme

Lignes 170-240 : Boucle pour calculer la valeur à comparer vg à partir de tous les termes de la formule de calcul en fonction des différentes valeurs de e (e est l'index de comptage de la boucle).
Si en ligne 230 la valeur à comparer est supérieure à 0, on saute au sous-programme (ligne 280) qui modifie les valeurs initiale et finale (qui constituent les limites de l'intervalle) ainsi que le pas de la boucle. On saute en outre dans ce cas à la ligne 170 (nouveau parcours de la boucle).

Lignes 250-260 : Sortie du résultat

Ligne 270 : Fin du programme

Lignes 280-300 : Sous-programme de réaffectation des variables k1, k2 et s. Les valeurs pour k1 et k2 sont choisies de façon à ce que la vraie valeur de e soit nécessairement comprise entre k1 et k2 (lors du parcours pour e-s, vg était encore négatif!).
En ligne 290 figure un test pour l'arrêt du programme.

Lignes 310-360 : Sous-programme "attendre"

Résultats du programme:

Si nous entrons après le lancement du programme les valeurs 50 (nombre d'échéances mensuelles), 0.5 (taux d'intérêt nominal par mois) et 1 (frais de gestion en %), nous obtenons le résultat suivant:

Taux d'intérêt annuel effectif:

e = 12.0329 %

Avec la formule du calcul mensuel nous obtiendrions un taux d'intérêt effectif légèrement plus élevé:

$$e = \frac{24 * (0.5 * 50 + 1)}{50 + 1} = 12.2353$$

4.5 Remboursement de dettes

Ce chapitre traite également des dettes. Si vous demandez par exemple un crédit de 100 000 F votre banque vous compte en général un agio. Cet agio doit être décompté mais il n'est pas payé. Il est simplement retenu par votre banque comme une sorte de taxe. Lors du calcul des intérêts et du remboursement de la dette vous devez donc partir de la dette nominale et non des sommes versées.

Lors de l'acceptation du crédit, vous convenez avec votre banque d'un taux d'intérêt et d'un remboursement initial (également en pourcentage du montant nominal du crédit). C'est de ces deux éléments que se composent les remboursements que vous aurez à effectuer chaque mois ou chaque trimestre. Avec un taux d'intérêt de 6% et un remboursement initial de 2%, cela représente 2000 F par trimestre ou 8000 F par an. Ces remboursements ne varient pas pendant toute la durée du crédit. La partie extinction de la dette de ces remboursements augmente toutefois constamment en même temps que diminue la partie correspondant aux intérêts puisque le capital produisant des intérêts se réduit d'année en année. C'est pourquoi le remboursement du prêt ne s'effectue pas dans l'exemple ci-dessus en 50 ans mais seulement en environ 23 ans (pour un taux d'intérêt inchangé).

Les offres de crédit différent notamment dans la durée du taux d'intérêt fixé conventionnellement. Une fois cette durée écoulée, le taux d'intérêt peut être augmenté ou diminué en fonction de l'état du marché. Mais vous pouvez alors dans ce cas rembourser immédiatement le solde de la dette si vous le souhaitez, que ce soit avec vos propres moyens ou que ce soit avec de l'argent à nouveau emprunté.

Il peut donc être très important pour vous de connaître le montant du reliquat de la dette. Ce montant peut être calculé à l'aide d'un plan de remboursement qui reconstitue la gestion de compte d'un institut de crédit. Il faut bien sûr prendre en compte tous les paramètres du crédit tels que remboursements, capital produisant des intérêts, intérêt, extinction de la dette et reliquat de la dette.
C'est un tel plan de remboursement que produit le programme suivant. On vous montre en même temps comment on peut avec le CPC créer des tableaux, avec le mode 80 caractères et l'instruction PRINT USING.

Quelques remarques sont cependant nécessaires au sujet de ce plan de remboursement:

1) On part comme base de calcul de la dette nominale et non du montant effectivement payé.

2) C'est cette dette nominale qui produit des intérêts et qui est remboursée

3) Bien que les remboursements soient payés régulièrement (tous les trimestres dans notre programme), l'intérêt est calculé pour chaque année pleine par rapport à un montant de capital inchangé. Ce n'est qu'après écoulement d'une année pleine que le capital produisant des intérêts diminue.

4) Le reliquat de la dette se réduit par contre avec chaque remboursement à raison du montant correspondant à l'extinction de la dette. Au bout d'une année pleine, ce reliquat est égal au capital produisant intérêt.

5) Les mouvements de compte sont calculés en fin d'année. Les intérêts, les paiements d'extinction de la dette et de remboursement sont additionnés et le capital est diminué du montant correspondant à l'extinction de la dette. Le reliquat de la dette n'est pas modifié par ces calculs puisqu'il diminue avec chaque remboursement (bien sûr uniquement lorsque cela correspond à l'extinction de la dette).

6) Le plan de remboursement se poursuit jusqu'à ce que le reliquat de la dette soit inférieur au remboursement ce qui est signalé par un message dans notre programme.

7) Pour optimiser la sortie à l'écran nous avons renoncé dans ce programme à sortir isolément le message "Frappez une touche S.V.P." lors de la création du tableau. Il faut par ailleurs tenir compte du nombre de chiffres dans l'instruction PRINT USING. Si vous entrez des montants de l'ordre du milliard, la sortie du tableau sera perturbée.

Programme:

```
10 REM I11
20 CLS
30 PRINT"I11 - Programme pour etablir"
40 PRINT TAB(7)"un plan de remboursement"
50 PRINT:PRINT:PRINT
60 PRINT TAB(11) "Bernd Kowal, 1985"
70 GOSUB 640
80 INPUT"Montant (nominal) du credit ";Kb:PRINT
90 INPUT"Taux d'interet nominal   ";Zs:PRINT
100 INPUT"Remboursement initial en % ";t:PRINT
110 PRINT"Entree de la date de versement :"
120 INPUT"Jour  ";t9
130 INPUT"Mois  ";m
140 INPUT"Annee ";j$:PRINT
150 j$=RIGHT$(j$,2):j=VAL(j$)
160 PRINT"C'est donc le";t9;".";m;".";j
170 PRINT"qui sera pris comme date de versement."
180 GOSUB 640:CLS:MODE 2
190 GOSUB 550
200 k1=1:k2=5:R=0:Zi=0:Ti=0:Rs=Kb
```

```
210 FOR i=k1 TO k2
220 IF i=k1 THEN 260
230 IF m>9 AND j=99 THEN j=0:GOTO 250
240 IF m>9 THEN j=j+1
250 IF m>9 THEN m=m-9 ELSE m=m+3
260 PRINT;USING"##.";t9;:PRINT;USING"##.";m;
270 PRINT;USING"##";j;
280 IF i=1 THEN PRINT TAB(11)"Versement";:GOTO 310
290 IF i=k1 THEN PRINT TAB(11)"Liquidation";:GOTO 310
300 PRINT TAB(11)"Traite ";i-1;
310 PRINT TAB(23);USING"#####.##";R;
320 PRINT;USING"    #######.##";Kb;
330 PRINT;USING"    #####.##";Zi;
340 PRINT;USING"    #####.##";Ti;
350 PRINT;USING"    #######.##";Rs
360 IF i>k1 THEN 410
370 IF i=1 THEN R=Kb*(Zs+t)/400 ELSE R=R/4
380 PRINT
390 Zi=Kb*Zs/400
400 Ti=R-Zi
410 Rs=Rs-Ti
420 IF Rs<0 THEN 510
430 IF i<k2 THEN 460
440 R=R*4:Zi=Zi*4:Ti=Ti*4
450 Kb=Kb-Ti:Rs=Rs+Ti/4
460 IF zae=3 AND k1=i THEN zae=0:GOSUB 640:GOSUB 560
470 NEXT i
480 zae=zae+1
490 k1=k2:k2=k2+4
500 GOTO 210
510 PRINT:PRINT"Avec le remboursement de";
520 PRINT INT((Rs+Ti)*100+0.5)/100;"FF, ";
530 PRINT"la dette est eteinte."
540 GOSUB 640:MODE 1:END
550 REM SP Titre du tableau
560 PRINT"Date         Operation     Traite     Capital";
570 PRINT"          Interet    Rembours.     Reste"
580 PRINT TAB(25)"en FF";TAB(35)"Produisant";
590 PRINT TAB(48)"en FF";TAB(58)"en FF";TAB(71)"du en FF
"
600 PRINT TAB(35)"interet"
610 PRINT TAB(35)"en FF"
620 PLOT 0,330:DRAW 620,330:PRINT
630 RETURN
640 REM SP Attendre
650 LOCATE 7,25
660 PRINT"Frappez une touche S.V.P"
670 x$=INKEY$:IF x$="" THEN 670
680 CLS:RETURN
```

Liste de variables:

i = index de comptage

j = année

j$ = tableau d'entrée pour j

k1 = valeur initiale de la boucle

k2 = valeur finale de la boucle

Kb = montant nominal du crédit, soit le capital produisant des intérêts

m = mois

R = remboursement en francs

Rs = reliquat de la dette en francs

t = remboursement initial en %

tg = jour

Ti = extinction de la dette en francs

zae = variable de comptage

Zi = intérêt en francs

Zs = taux d'intérêt nominal en %

Description du programme:

Lignes 10-70 : Titre

Lignes 80-170 : Entrée des valeurs initiales
L'indication de l'année est traitée comme une variable alphanumérique (=chaîne de caractères) de façon à pouvoir en ligne 150 ne convertir que les deux derniers caractères en une expression numérique. Cette méthode un peu compliquée a pour effet de n'utiliser que "85" comme année lors de la création du tableau si l'entrée était par exemple "1985".

Ligne 180 : On saute au sous-programme "attendre", l'écran est vidé et on change le mode écran (80 colonnes)

Ligne 190 : Saut au sous-programme (ligne 550) pour sortir le titre du tableau

Ligne 200 : Affectation des valeurs initiales aux variables pour le premier parcours de la boucle

Lignes 210-470 : Constitution d'une boucle de calcul des différentes grandeurs et de sortie des résultats sous forme de tableau pour une période (=1 an). Lors de chaque premier parcours on prend en compte l'opération "Paiement" ou "Solde".
220-250 : Pour les parcours 2 à 5 de chaque période (c'est alors que les remboursements sont payés), modification de la date
260-270 : Sortie de la date

280-300 : Sortie de l'opération dont il s'agit (1er parcours de la première période: paiement, 1er parcours des périodes suivantes: solde, parcours 2 à 5 de toutes les périodes: remboursement)

310-350 : Sortie des autres valeurs du tableau

360-400 : Lors de chaque premier parcours, calcul des remboursement, intérêts et extinction de la dette (montants constants pour chaque période) et sortie d'une ligne vide. A noter en ligne 370 que le remboursement n'est en fait calculé que lors du premier parcours de la première période. Pour le reste, le dernier solde est simplement divisé par 4.

430-450 : Lors de chaque dernier parcours, sont calculésles remboursement, intérêt, capital produisant intérêts et reliquat de la dette pour le solde.

460 : Sortie des résultats pour 3 périodes (zae=3), y compris solde (kl=i, premier parcours), remise à 0 de l'index de comptage i et saut aux sous-programme "attendre" et "titre du tableau".

Lignes 480-500 : A la fin du parcours de la boucle pour une période, l'index de comptage est augmenté de 1, les valeurs initiale et finale de la boucle reçoivent de nouvelles valeurs et on saute à la ligne 210 (nouveau parcours de la boucle).

Lignes 510-530 : Sortie d'un commentaire concernant le reliquat de la dette lorsqu'il ne reste plus de remboursement à effectuer.

Ligne 540 : Saut au sous-programme "attendre", passage en mode 40 colonnes et fin du programme

Lignes 550-630 : Sous-programme "titre du tableau"
Le titre est souligné en ligne 620. Le curseur graphique est amené avec l'instruction PLOT dans l'emplacement voulu et la ligne est dessinée avec l'instruction DRAW.

Lignes 640-690 : Sous-programme "attendre"
Le commentaire "Frappez une touche S.V.P." n'est sorti que si la création du tableau n'a pas encore commencé.

Résultats du programme:

Lancez le programme et entrez un montant de crédit nominal de 10000 F, un intérêt annuel de 7% et un remboursement initial de 3% avec versement de l'argent le 30 12 1985. Vous verrez alors que pour un versement trimestriel de 225 F, vous devrez verser 88 remboursements jusqu'au 30 12 07 (donc exactement 22 ans) et qu'il faut en outre payer un reliquat de 198.85 F. Vous rembourserez donc au total 19998.85.

4.6 Le puits à idées

- mise en place d'un compte comparatif pour comparer les offres de crédit (ce sont ici - contrairement au plan de remboursement - les paiements effectifs aux échéances effectives qui jouent un rôle déterminant).

- Détermination du montant évalué pour les contrats d'épargne-logement.

- Financement immobilier et modèles de financement de promoteurs.

- Gestion et calcul des coûts.

- Analyse boursière: analyse de cours et des tendances de la bourse.

- Gestion d'un portefeuille d'actions.

- et bien d'autres choses.

5. Textes et impression

5.1 Remarque préalable

Un microordinateur ne sert pas qu'à calculer mais aussi à gérer des textes. Cette application de l'informatique est certainement connue même de ceux qui ne sont pas employés de bureau.
Les tâches principales du traitement de texte sont:

- sauvegarde et archivage de textes

- sortie répétée de textes

- modification rapide et à volonté de textes (correction, complément, fusion, etc.).

Si vous voulez vraiment vous adonner sérieusement au traitement de texte, nous vous conseillons de vous intéresser au marché des logiciels professionnels. Vous serez surpris de voir par exemple tout ce que permet de faire TEXTOMAT de DATA BECKER - MICRO APPLICATION. Il faut cependant pour cela posséder un lecteur de disquette comme moyen de stockage externe. Pour une utilisation domestique occasionnelle, vous pouvez cependant vous contenter de programmes que vous aurez faits vous-même et d'un lecteur de cassette. Un tel programme peut soit viser à imiter les fonctions les plus importantes des logiciels de traitement de texte professionnels (voir chapitre 5.4) soit être approprié uniquement à certaines applications déterminées (voir chapitre 5.3).

Il n'est par ailleurs pas très difficile d'affecter les caractères spéciaux français comme par exemple à, ç, é, è ou ù à certaines touches de votre ordinateur. C'est très important pour un traitement de texte. Vous trouverez au chapitre 5.2 un programme remplissant cette fonction.

On ne peut faire de recommandations générales en ce qui concerne les imprimantes. L'imprimante matricielle AMSTRAD NLQ 401 offre certainement une belle écriture pour son prix mais quelle que soit l'imprimante que vous possédez, vous en trouverez toujours une qui ait une plus belle qualité d'impression. Le choix entre une imprimante à marguerite et une imprimante matricielle est notamment essentiellement un problème de moyens mais l'imprimante à marguerite est bien sûr plus appropriée à de véritables applications professionnelles.

Un dernier conseil au sujet du moniteur. Si vous avez un moniteur couleur, essayez de trouver avec quelle combinaison de couleurs vous pouvez le mieux travailler. Evitez absolument de rechercher des couleurs vives. Vous yeux vous remercieront. Pour un travail nécessitant une lecture attentive et prolongée de l'écran, le mieux est un moniteur monocolore vert. Il convient parfaitement au traitement de texte car il ne distrait pas de l'essentiel.

5.2 Caractères spéciaux français

Si vous comparez le clavier de votre ordinateur avec celui d'une machine à écrire, vous voyez tout de suite la différence dans l'emplacement des touches. D'autre part, l'absence des accents français est difficilement acceptable pour le traitement de texte. Le BASIC AMSTRAD permet heureusement très facilement de définir ces caractères spéciaux et de les affecter à des touches déterminées. Il vaudrait cependant mieux veiller à ne choisir comme touches que celles dont le code ASCII correspond à la valeur du caractère spécial dans la table de codes de caractère de l'imprimante. On est ainsi sûr qu'un accent à l'écran soit bien imprimé comme accent.

Le programme présenté ici réalise l'affectation des caractères spéciaux aux touches correspondantes. Le maniement de ces touches nécessite cependant une phase d'adaptation car les touches sélectionnées ne correspondent pas toujours à celles d'une machine à écrire (clavier AZERTY).

La réaffectation des touches se fait avec l'instruction SYMBOL. Il faut noter à cet égard que les paramètres de cette instruction qui décrivent bit par bit en commençant par le haut la matrice de 8 points sur 8 du nouveau caractère sont des nombres hexadécimaux. Le paramètre C3 a par exemple pour effet que les deux premières et les deux dernières colonnes de la ligne concernée reçoivent la couleur d'écriture alors que les quatre colonnes du milieu recevront la couleur du fond. En effet C3 s'écrit en binaire:

11000011

0 = couleur du fond
1 = couleur d'écriture

Pour chacune des huit lignes vous devez entrer un nombre hexadécimal correct. Le manuel d'utilisation de votre CPC contient une belle table de conversion binaire - hexadécimal. Vous pouvez ainsi vous éviter des calculs fastidieux. Le travail de définition des caractères est d'autre part considérablement allégé si vous vous aidez du jeu de caractères du CPC qui est décrit dans votre manuel. Vous trouvez également au même endroit les numéros de caractère correspondants (premier paramètre de l'instruction SYMBOL).

Programme:

```
10 REM I12
20 CLS
30 PRINT "I12 - Programme de definition des"
40 PRINT TAB(7)"caracteres francais speciaux"
50 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
60 PRINT TAB(9) "Bernd Kowal, 1985"
70 GOSUB 350
80 PRINT"Veuillez actionner lentement l'une"
90 PRINT"apres l'autre les touches suivantes :",
100 PRINT:PRINT"sans SHIFT     : @, \, [, ]"
110 PRINT:PRINT"et avec SHIFT : |, {, }"
120 SYMBOL AFTER 63
130 REM a accent grave
140 SYMBOL 64,&60,&30,&78,&C,&7C,&CC,&76,&0
150 REM degre
160 SYMBOL 91,&18,&24,&24,&18,&0,&0,&0,&0
170 REM c cedille
180 SYMBOL 92,&0,&0,&3C,&66,&60,&66,&3C,&18
190 REM paragraphe
200 SYMBOL 93,&1E,&30,&38,&6C,&38,&18,&F0,&0
210 REM e accent aigu
220 SYMBOL 123,&C,&18,&3C,&66,&7E,&60,&3C,&0
230 REM u accent grave
240 SYMBOL 124,&30,&18,&66,&66,&66,&66,&3F,&0
250 REM e accent grave
260 SYMBOL 125,&30,&18,&3C,&66,&7E,&60,&3C,&0
270 PRINT:PRINT:PRINT
280 PRINT"Si le dessin des accents ne vous"
290 PRINT"satisfait pas, il vous suffit de"
300 PRINT"modifier les instructions SYMBOL"
310 PRINT"correspondantes."
320 PRINT:PRINT:PRINT
330 PRINT"Fin du programme":END
340 REM SP Attendre
350 LOCATE 7,25
360 PRINT"Frappez une touche S.V.P"
370 x$=INKEY$:IF x$="" THEN 370
380 CLS:RETURN
```

Le programme n'utilise pas de variables

Description du programme:

Lignes 10-70 : Titre

Lignes 80-120 : Sortie d'une instruction pour l'utilisateur

Ligne 130 : Fixation de la zone dans laquelle doivent être définis les caractères spéciaux français

Lignes 140-270 : Définition des caractères spéciaux français et réaffectation des touches

Lignes 280-330 : Sortie d'un commentaire

Ligne 340 : Fin du programme

Lignes 350-400 : Sous-programme "attendre"

Résultats du programme:

Vous pouvez utiliser ces instructions SYMBOL chaque fois que vous avez besoin des caractères spéciaux français. Cela peut se faire dans le cours du programme, comme dans notre programme, ou en mode direct, en entrant directement ces instructions.

5.3 Une lettre d'affaires individualisée

Vous êtes commerçant et vous possédez un petit magasin joliment appelé "Commerce ONVANTOU" 1, rue du hameau à Trifouilly. Il peut malheureusement arriver qu'un client "oublie" de régler une facture. Il vous faut alors envoyer à ce cher client (que nous appellerons Monsieur Avare) une mise en demeure qui pourrait se présenter ainsi:

(voir page suivante)

Commerce ONVANTOU
1, rue du hameau

99999 Trifouilly

Richard Avare
2, rue de la ville

99999 Trifouilly   Trifouilly, le 20.4.1985

Objet: Livraison d'un tiroir du 1 4 1985

Cher Monsieur Avare,

Vous n'avez pas encore réglé la facture indiquée ci-dessus. C'est inadmissible. Si le montant de 147,50 F n'est pas viré sur notre compte avant le 30 4 1985, nous entamerons une procédure contentieuse.

Salutations distinguées

Si maintenant un autre client se fait également attendre pour le règlement d'une facture, il suffira de modifier uniquement les parties soulignées de la lettre.

C'est exactement ainsi que vous pouvez procéder avec votre CPC, une imprimante et un programme approprié. Vous trouverez dans les pages suivantes le programme correspondant à la lettre commerciale ci-dessus. Les parties soulignées de la lettre peuvent être entrées à travers une instruction INPUT. Le reste de la lettre est par contre définitivement fixé dans le programme. Si le texte de la lettre ne vous convient pas (les pratiques commerciales à Paris sont différentes de ce qu'elles sont à Trifouilly) vous n'avez qu'à le modifier. Vous n'aurez de toute façon à faire ce travail de modification qu'une seule fois. Notez cependant que par exemple dans notre programme, chaque ligne de texte compte exactement 56 caractères (y compris les espaces). Cela fait un effet professionnel et c'est très simple à programmer avec des espaces supplémentaires.

Vous pouvez ainsi traiter toute votre correspondance commerciale. Vous serez surpris de constater combien de sections de texte se répètent dans votre correspondance. Et cependant le caractère individualisé de vos lettres commerciales est préservé.

Programme:

```
10 REM I13
20 CLS
30 PRINT "I13 - Programme de redaction d'une"
40 PRINT TAB(7)"lettre d'affaire individualis{e"
50 PRINT TAB(7)"(Mise en demeure)"
60 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
70 PRINT TAB(9) "Bernd Kowal, 1985"
80 GOSUB 680
90 REM caract}res fran\ais sp{ciaux
100 SYMBOL AFTER 63
110 SYMBOL 64,&60,&30,&78,&C,&7C,&CC,&76,&0
120 SYMBOL 91,&18,&24,&24,&18,&0,&0,&0,&0
130 SYMBOL 92,&0,&0,&3C,&66,&60,&66,&3C,&18
140 SYMBOL 93,&1E,&30,&38,&6C,&38,&18,&F0,&0
150 SYMBOL 123,&C,&18,&3C,&66,&7E,&60,&3C,&0
160 SYMBOL 124,&30,&18,&66,&66,&66,&66,&3F,&0
170 SYMBOL 125,&30,&18,&3C,&66,&7E,&60,&3C,&0
180 REM Entr{es
190 MODE 2
200 PRINT"Nom du client :"
210 INPUT"Pr{nom    ";vn$
220 INPUT"Nom       ";nn$
230 PRINT:PRINT"Le client est-il un homme ou une femme?"
240 INPUT"(entrez m ou f) ";g$
250 PRINT:PRINT"Adresse du client:"
260 INPUT"Rue et num{ro ";sh$
270 INPUT"Ville et code postal   ";op$
280 PRINT:INPUT"Date d'exp{dition ";ad$
290 PRINT:INPUT"Objet ";b$
300 PRINT:INPUT"Montant concern{ en FF ";dm
310 PRINT:PRINT"Jusqu'@ quand les paiements sont-ils"
320 INPUT"pris en compte ";zd$
330 REM Sortie
340 CLS:PRINT"Sortie sur";TAB(30)"Entr{e"
350 PRINT:PRINT"Ecran";TAB(33)"0"
360 PRINT:PRINT"Imprimante";TAB(33)"8"
370 PRINT:PRINT:INPUT "Votre choix ";a:CLS
380 PRINT #a:PRINT #a:PRINT #a:PRINT #a
390 PRINT #a, "Commerce ONVANTOU"
400 PRINT #a, "1, rue du Hameau"
410 PRINT #a:PRINT #a, "99999 Trifouilly"
420 PRINT #a:PRINT #a:PRINT #a
430 PRINT #a, vn$;" ";nn$
440 PRINT #a, sh$
450 PRINT #a:PRINT #a, op$;
460 q=57-21-LEN(ad$)
470 PRINT #a, TAB(q)"Trifouilly, le ";ad$
```

```
480 PRINT #a:PRINT #a:PRINT #a:PRINT #a:PRINT #a
490 PRINT #a, "Objet: ";b$
500 PRINT #a:PRINT #a
510 IF a=0 THEN GOSUB 680
520 IF g$="m" THEN 540
530 PRINT #a,"Ch}re Madame ";nn$;",":GOTO 550
540 PRINT #a,"Cher Monsieur ";nn$;","
550 PRINT #a:PRINT #a, "Vous n'avez pas encore r{gl{ ";
560 PRINT #a, "la facture indiqu{e ci-"
570 PRINT #a, "dessus. C'est inadmissible ! Si la somme
de ";
580 PRINT #a, USING "####.##";dm;
585 PRINT#a, " F"
590 PRINT #a, "n'est pas vir{e sur notre compte avant le
 ";
600 PRINT #a, USING"\          \";zd$;
610 PRINT #a,", nous"
620 PRINT #a, "entamerons une proc{dure contentieuse."
630 PRINT #a:PRINT #a:PRINT #a:PRINT #a:PRINT #a
640 PRINT #a,"Salutations distingu{es"
650 PRINT #a:PRINT #a:PRINT #a:PRINT #a:PRINT #a
660 GOSUB 680:MODE 1:END
670 REM SP Attendre
680 LOCATE 7,25
690 PRINT"Frappez une touche S.V.P"
700 x$=INKEY$:IF x$="" THEN 700
710 CLS:RETURN
```

Liste de variables:

a = périphérique de sortie (0 pour l'écran, 8 pour l'imprimante)

ad$ = date d'expédition de la lettre

b$ = objet

dm = somme demandée en francs

g$ = sexe du client

nn$ = nom de famille du client

op$ = localité avec code postal

q = paramètre calculé pour la fonction TAB pour la sortie de la date d'expédition

sh$ = rue avec numéro

v$ = prénom du client

zd$ = date jusqu'à laquelle les paiements ont été pris en compte

Description du programme:

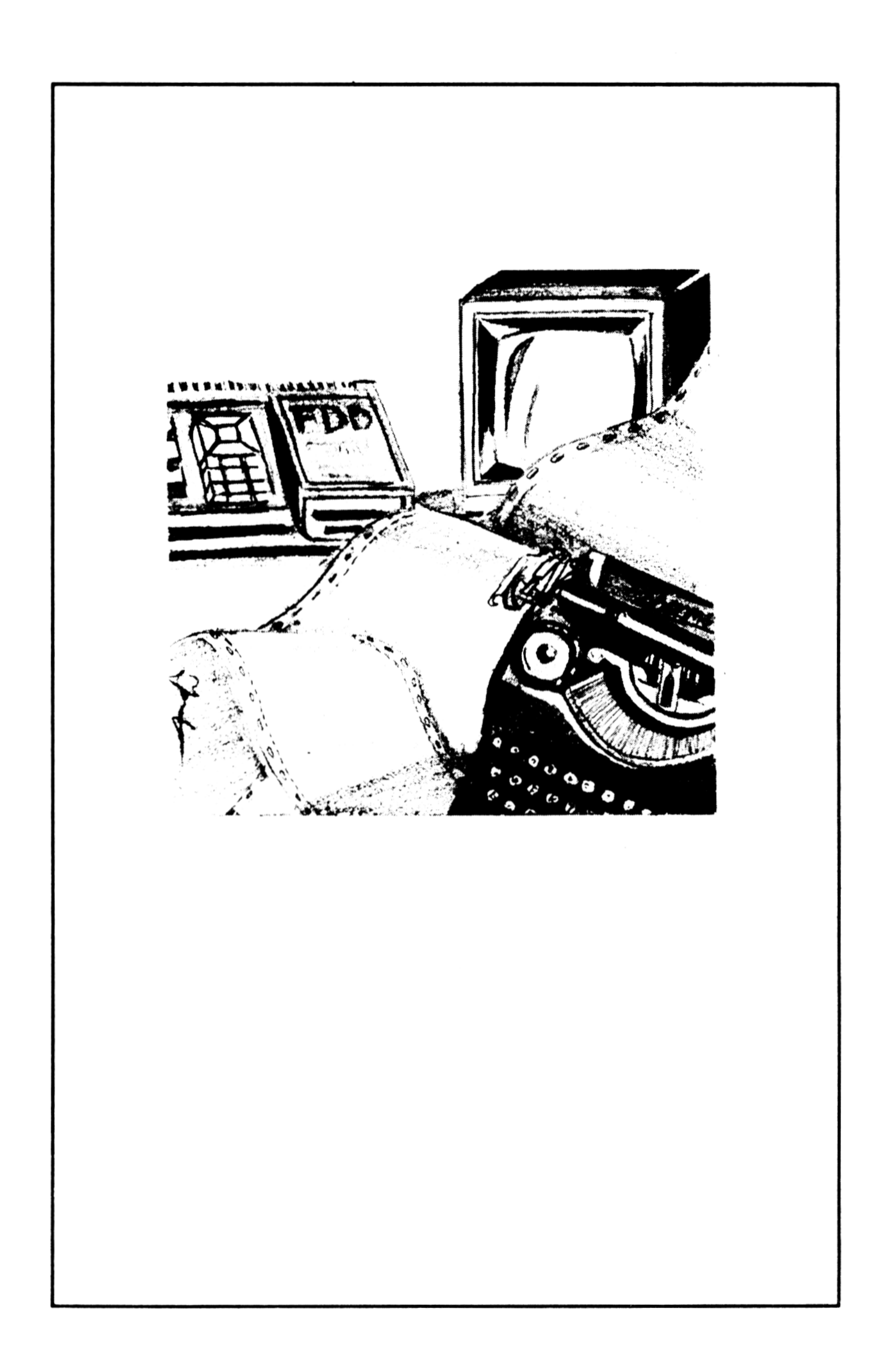
Lignes 10-80 :	Titre
Lignes 90-170 :	Définition des accents français (voir chapitre 5.2)

Lignes 180-320 : Entrée des parties "soulignées" de la lettre. Le mode écran est fixé à cet effet à 80 colonnes (pour admettre des entrées plus longues).

Lignes 330-660 : Sortie de la lettre complète
Vous pouvez choisir en lignes 340-370 si la lettre doit être sortie sur l'écran ou sur l'imprimante. Notez qu'une mauvaise entrée pour a ne serait pas "interceptée" par le programme. En ligne 460, le paramètre de la fonction TAB est calculé pour la sortie en ligne 470 de la date d'expédition, de façon à ce que la ligne sortie compte exactement 56 caractères. Dans le cas d'une sortie à l'écran, le déroulement du programme est interrompu en ligne 510 (Sous-programme "attendre") car la lettre compte plus de lignes que l'écran. En lignes 520-540, la formule d'adresse (Cher Monsieur ou Chère Madame) est sortie en fonction de l'entrée "sexe du client".
La sortie formatée du montant demandé en francs et de la date se fait en lignes 580-600. Si vous avez affaire à des montants plus élevés, vous devez changer le format en tenant compte du fait que la ligne correspondante comptera alors plus de caractères. Pour la date, un maximum de 10 caractères a été prévu, ce qui suffit par exemple pour une entrée telle que "24 12 1985".

Ligne 670 : Saut au sous-programme "attendre", modification du mode écran et fin du programme

Lignes 680-730 : Sous-programme "attendre"


FDD

5.4 Traitement de texte

Comparé à des logiciels de traitement de texte professionnels, il est évident que le programme que nous vous proposons dans ce chapitre présente des insuffisances. Mais après tout, il a été conçu uniquement pour un usage domestique et non pour travailler au bureau.

Ce programme vous permet d'effectuer les tâches suivantes:

- Entrée de textes (au clavier)
- Examen de textes (à l'écran)
- Correction de textes (correction, insertion ou suppression de lignes de texte).
- Chargement de textes (de la cassette ou de la disquette)
- Sauvegarde de textes (sur cassette ou sur disquette)
- Impression de textes

Si vous voulez employer un lecteur de disquette comme moyen de stockage externe, vous devez modifier ou insérer les lignes de programme suivantes:

```
1745 INPUT"Nom du texte ";n$
1750 OPENIN n$
```

Aucune autre modification n'est nécessaire.

Vous pouvez choisir vous-même le nombre de caractères par ligne (jusqu'à 70). Vous pouvez également déterminer librement le nombre de lignes par page et la largeur de la marge gauche lors de la sortie sur imprimante. Aucune autre indication de format n'est employée. Le texte est imprimé exactement comme vous l'entrez et comme il apparaît à l'écran.

Contrairement à ce qui est le cas dans notre programme, le formatage, c'est-à-dire la préparation d'un texte pour l'impression, se fait dans beaucoup de programmes de traitement de texte dans une étape de travail particulière. Le moniteur n'est pas en effet en mesure de représenter tout ce que l'imprimante peut sortir. D'autre part, dans beaucoup de programmes de traitement de texte, on n'a pas à surveiller la longueur d'une ligne de texte. Vous pouvez en effet indiquer la longueur de ligne voulue simplement avant la sortie sur imprimante. Vous pouvez naturellement également d'abord faire sortir le texte formaté à l'écran. On peut dire en conclusion que les programmes professionnels vous offrent, outre une manipulation plus aisée du texte, de nombreuses possibilités de sortir sur papier un texte déjà entré en mémoire.

Il en va quelque peu différemment pour le programme que nous vous proposons ici. Le nombre de caractères par ligne est également limité lors de la sortie sur imprimante mais vous devez vous-même veiller lors de l'entrée du texte à ne pas entrer trop de caractères par ligne. Le programme dessine simplement une ligne verticale à l'écran qui vous indique quand la longueur limite d'une ligne est atteinte. C'est toujours le cas lorsque le curseur touche cette ligne. Il vous faut alors appuyer sur la touche ENTER avant d'écrire la ligne suivante. Pendant l'entrée du texte, un affichage vous indique sur quelle ligne et dans quelle page vous vous trouvez. Il faut par ailleurs noter dans ce programme que les guillemets doivent être remplacés par l'apostrophe pour ne pas perdre la suite de la ligne.

Un texte traité avec le programme que nous vous proposons peut contenir jusqu'à un maximum de 300 lignes de texte (indépendamment du découpage en pages). Cela représente environ 6 pages de machine à écrire. Si vous modifiez les lignes 180 et 570, vous pouvez même entrer un nombre plus important de lignes de texte. Vous devez simplement tenir compte de la capacité mémoire de votre ordinateur. Vous pouvez à cet effet interrompre le déroulement du programme pour contrôler avec la fonction FRE la place mémoire encore libre ou bien vous pouvez également intégrer l'interrogation de cette fonction dans votre programme pour empêcher une saturation de la mémoire.

Si vous voulez compléter un texte déjà sauvegardé, vous pouvez le charger en mémoire et y insérer alors de nouvelles lignes de texte. Les indications de format "caractères par ligne" et "lignes par page" doivent cependant être conservées inchangées. Seule l'indication "lignes par page" peut être encore modifiée lors de la sortie sur imprimante.

Il n'est pas possible de charger un texte à la suite d'un texte déjà en mémoire. Le texte en mémoire serait en effet effacé. Par contre on peut compléter un texte chargé en mémoire en n'importe quel endroit de ce texte.

Pour le reste, ce programme est largement auto-explicatif. Le mieux est que vous l'essayiez directement. Et surtout n'éteignez pas votre ordinateur avant d'avoir sauvegardé votre texte.

Programme:

```
10 REM I14
20 CLS
30 PRINT "I14 - Programme de traitement de texte"
40 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
50 PRINT TAB(9) "Bernd Kowal, 1985"
60 GOSUB 2490
70 REM
80 REM caract}res fran\ais sp{ciaux
90 REM
100 SYMBOL AFTER 63
110 SYMBOL 64,&60,&30,&78,&C,&7C,&CC,&76,&0
120 SYMBOL 91,&18,&24,&24,&18,&0,&0,&0,&0
130 SYMBOL 92,&0,&0,&3C,&66,&60,&66,&3C,&18
140 SYMBOL 93,&1E,&30,&38,&6C,&38,&18,&F0,&0
150 SYMBOL 123,&C,&18,&3C,&66,&7E,&60,&3C,&0
160 SYMBOL 124,&30,&18,&66,&66,&66,&66,&3F,&0
170 SYMBOL 125,&30,&18,&3C,&66,&7E,&60,&3C,&0
180 DIM t$(300)
190 OPENOUT "d":MEMORY HIMEM-1:CLOSEOUT
200 REM
210 REM Menu
220 REM
230 PRINT TAB(15)"Menu":PRINT
240 PRINT TAB(33)"Entr{e":PRINT
250 PRINT
260 PRINT" entrer texte";TAB(36)"1":PRINT
270 PRINT" examiner texte";TAB(36)"2":PRINT
280 PRINT" corriger texte";TAB(36)"3":PRINT
290 PRINT" charger texte";TAB(36)"4":PRINT
300 PRINT" sauvegarder texte";TAB(36)"5":PRINT
310 PRINT" imprimer texte";TAB(36)"6":PRINT
320 PRINT"Fin du travail";TAB(36)"7"
330 PRINT:PRINT:PRINT:INPUT"Votre choix ";a:CLS
340 IF a<1 OR a>7 THEN 230
350 ON a GOSUB 380,650,850,1730,1830,1950,2260
360 MODE 1:GOTO 230
370 REM
380 REM Entrer textes
390 REM
400 IF zz>0 THEN 430
410 INPUT"Combien de colonnes par ligne ";zz:PRINT
420 IF zz<1 OR zz>70 THEN 410
430 IF zs>0 THEN 470
440 INPUT"Combien de lignes par page    ";zs:PRINT
450 PRINT:PRINT"Si vous entrez '##' pour une ligne,"
460 PRINT"vous mettez fin @ l'entr{e.":GOSUB 2490
470 GOSUB 2370
480 WINDOW #0,1,7+zz,4,25
```

```
490 IF s=0 THEN s=1
500 IF z<1 THEN 550
510 k1=s*zs-zs+1:k2=s*zs-zs+z
520 FOR i=k1 TO k2
530 PRINT #0, USING " ##    ";i-s*zs+zs;
540 PRINT #0, t$(i):NEXT i
550 IF z=zs THEN z=0:s=s+1
560 z=z+1:az=az+1
570 IF az=300 THEN PRINT:PRINT"Il n'y a plus de place di
sponible en m{moire":RETURN
580 LOCATE #1, zz/2+7,1:PRINT #1, s
590 PRINT #0, USING " ##    ";z;
600 LINE INPUT t$(az)
610 IF t$(az)<>"##" THEN 550
620 t$(az)=""
630 z=z-1:az=az-1:WINDOW #0,1,40,1,25:RETURN
640 REM
650 REM Examen des textes
660 REM
670 INPUT"Quelle page ";x
680 IF x=0 THEN 670
690 l=0:GOSUB 2370
700 WINDOW #0,1,7+zz,4,25
710 LOCATE #1, zz/2+7,1:PRINT #1, x
720 k1=x*zs-zs+1:k2=x*zs
730 FOR i=k1 TO k2
740 l=l+1
750 PRINT #0, USING " ##    ";i-x*zs+zs;
760 PRINT #0, t$(i)
770 IF l=22 THEN l=0:GOSUB 2540
780 NEXT i:GOSUB 2540:MODE 1
790 WINDOW #0,1,40,1,25
800 INPUT"Encore une page (o/n) ";a$
810 IF a$="n" THEN RETURN
820 IF a$<>"o" THEN 800
830 PRINT:PRINT:GOTO 670
840 REM
850 REM Correction de textes
860 REM
870 PRINT TAB(15)"Menu":PRINT:PRINT
880 PRINT TAB(30)"Entr{e":PRINT:PRINT
890 PRINT"Travail sur des lignes de texte:":PRINT:PRINT
900 PRINT" correction";TAB(33)"1":PRINT
910 PRINT" insertion";TAB(33)"2":PRINT
920 PRINT" suppression";TAB(33)"3":PRINT:PRINT
930 PRINT"Fin de là correction";TAB(33)"4":PRINT
940 PRINT:PRINT:INPUT"Votre choix ";a:MODE 2
950 IF a<1 OR a>4 THEN 870
```

```
960 IF a<4 THEN INPUT"Quelle page ";x:PRINT
970 ON a GOTO 990,1150,1470,1700
980 REM
990 REM corriger
1000 REM
1010 PRINT"Quelle ligne doit"
1020 INPUT"etre modifi{e ";y
1030 i=(x-1)*zs+y
1040 PRINT:PRINT"La ligne pr{c{dente :"
1050 PRINT t$(i-1);:PRINT TAB(zz+1) CHR$(211):PRINT
1060 PRINT"La ligne @ corriger :"
1070 PRINT t$(i);:PRINT TAB(zz+1) CHR$(211):PRINT
1080 PRINT"La ligne suivante :"
1090 PRINT t$(i+1);:PRINT TAB(zz+1) CHR$(211):PRINT
1100 PRINT:PRINT"Comment doit se pr{senter la ligne";y;"
?"
1110 LOCATE zz+1,17:PRINT CHR$(211):PRINT
1120 LOCATE 1,17:LINE INPUT t$(i)
1130 MODE 1:GOTO 870
1140 REM
1150 REM insertion
1160 REM
1170 PRINT"Entre quelles lignes faut-il placer"
1180 PRINT"l'insertion ?"
1190 PRINT:INPUT"Num{ro de ligne inf{rieur  ";y1
1200 z1=y1:y1=(x-1)*zs+y1
1210 PRINT"La ligne se pr{sente ainsi :"
1220 PRINT t$(y1);:PRINT TAB(zz+1) CHR$(211)
1230 PRINT:INPUT"Num{ro de ligne sup{rieur  ";y2
1240 IF y2-z1=1 THEN 1270
1250 PRINT:PRINT"Vous devez indiquer deux num{ros de"
1260 PRINT"ligne cons{cutifs.":GOTO 1170
1270 y2=(x-1)*zs+y2
1280 PRINT"La ligne se pr{sente ainsi :"
1290 PRINT t$(y2);:PRINT TAB(zz+1) CHR$(211)
1300 PRINT:PRINT:PRINT"Combien de lignes faut-il"
1310 INPUT"ins{rer ";y3
1320 az=az+y3:z=z+y3
1330 IF z>zs THEN z=z-zs:s=s+1
1340 FOR i=az TO y2 STEP -1
1350 t$(i+y3)=t$(i)
1360 NEXT i:GOSUB 2370
1370 WINDOW #0,1,7+zz,4,25
1380 FOR i=y1+1 TO y1+y3
1390 z1=z1+1
1400 LOCATE #1, zz/2+7,1:PRINT #1, x
1410 PRINT #0, USING " ##   ";z1;
1420 LINE INPUT t$(i)
```

```
1430 NEXT i
1440 WINDOW #0,1,40,1,25
1450 MODE 1:GOTO 870
1460 REM
1470 REM supprimer
1480 REM
1490 PRINT"Quelles lignes faut-il supprimer?
1500 PRINT:INPUT"de: num{ro de ligne ";y1
1510 z1=y1:y1=(x-1)*zs+y1
1520 PRINT"La ligne se pr{sente ainsi :"
1530 PRINT t$(y1);:PRINT TAB(zz+1) CHR$(211)
1540 PRINT:INPUT"jusqu'@: num{ro de ligne ";y2
1550 IF y2>=z1 THEN 1580
1560 PRINT:PRINT"Le deuxi}me num{ro de ligne doit etre"
1570 PRINT"plus grand que le premier.":GOTO 1500
1580 y2=(x-1)*zs+y2
1590 PRINT"La ligne se pr{sente ainsi :"
1600 PRINT t$(y2);:PRINT TAB(zz+1) CHR$(211)
1610 PRINT:PRINT:PRINT TAB(10)"Je supprime !"
1620 GOSUB 2490
1630 y3=y2-y1+1
1640 FOR i=y1 TO az
1650 t$(i)=t$(i+y3)
1660 NEXT i
1670 az=az-y3:z=z-y3
1680 IF z<1 THEN z=z+zs:s=s-1
1690 MODE 1:GOTO 870
1700 REM Fin de la correction
1710 MODE 1:RETURN
1720 REM
1730 REM Charger textes
1740 REM
1750 OPENIN ""
1760 INPUT #9,zz,zs,s,z,az
1770 FOR i=1 TO az
1780 INPUT #9,t$(i)
1790 NEXT i
1800 CLOSEIN
1810 CLS:RETURN
1820 REM
1830 REM Sauvegarder textes
1840 REM
1850 PRINT"Comment doit s'appeler le texte"
1860 INPUT"@ sauvegarder ";n$:PRINT
1870 OPENOUT n$
1880 PRINT #9,zz,zs,s,z,az
1890 FOR i=1 TO az
1900 WRITE #9,t$(i)
```

```
1910 NEXT i
1920 CLOSEOUT
1930 CLS:RETURN
1940 REM
1950 REM imprimer des textes
1960 REM
1970 PRINT"Faut-il modifier le nombre de lignes"
1980 PRINT"par page (o/n) ";
1990 INPUT a$
2000 IF a$="n" THEN 2030
2010 IF a$<>"o" THEN 1970
2020 PRINT:INPUT"Nouvelle valeur";zs
2030 PRINT:PRINT"Largeur de la marge gauche ?"
2040 INPUT"Nombre d'espaces ";lr
2050 WIDTH zz+lr
2060 PRINT:PRINT"Faut-il imprimer la totalit{"
2070 INPUT"du texte (o/n) ";a$
2080 IF a$="n" THEN 2140
2090 IF a$<>"o" THEN 2060
2100 FOR i=1 TO az
2110 PRINT #8, TAB(lr+1) t$(i)
2120 NEXT i
2130 GOTO 2240
2140 PRINT:PRINT"Quelle page faut-il"
2150 INPUT"imprimer ";x
2160 k1=x*zs-zs+1:k2=x*zs
2170 FOR i=k1 TO k2
2180 PRINT #8, TAB(lr+1) t$(i)
2190 NEXT i
2200 PRINT:PRINT"Faut-il imprimer une autre page"
2210 INPUT"de texte (o/n) ";a$
2220 IF a$="o" THEN 2140
2230 IF a$<>"n" THEN 2200
2240 CLS:RETURN
2250 REM
2260 REM Fin du travail
2270 REM
2280 PRINT"N'avez-vous pas oubli{ de sauvegarder"
2290 PRINT"le texte ?"
2300 PRINT"Si c'est le cas, entrez simplement"
2310 PRINT"un 'oh'. Sinon, il suffit d'appuyer"
2320 PRINT"sur la touche espace.":INPUT a$
2330 IF a$="oh" THEN GOSUB 1830
2340 PRINT:PRINT:PRINT
2350 PRINT"Vous pouvez maintenant me d{brancher."
2360 PRINT:PRINT:PRINT TAB(8)"Au revoir":END
2370 REM
2380 REM Sous-programme masque {cran
```

```
2390 REM
2400 MODE 2
2410 WINDOW #1,1,80,1,3
2420 PRINT #1,TAB(zz/2)"Page"
2430 PRINT #1,"Ligne";
2440 WINDOW #2,8+zz,9+zz,4,25
2450 FOR i=1 TO 25
2460 PRINT #2, CHR$(211):NEXT i
2470 RETURN
2480 REM
2490 REM SP Attendre
2500 REM
2510 LOCATE 7,25
2520 PRINT"Frappez une touche S.V.P"
2530 REM
2540 x$=INKEY$:IF x$="" THEN 2540
2550 CLS:RETURN
```

Liste de variables:

a = réponse au menu

a$ = chaîne de réponse

az = nombre de lignes absolu ou numéro de ligne (se rapportant à toutes les lignes de texte figurant en mémoire)

i = index de comptage et numéro de ligne absolu dans la section de programme "correction"

k1 = limite inférieure d'une boucle de programme

k2 = limite supérieure d'une boucle de programme

l = grandeur auxiliaire dans la section de programme "examen du texte"

lr = largeur de la marge gauche lors de la sortie sur imprimante

n$ = nom du texte sauvegardé

s = nombre de pages

t$(i) = ligne de texte

x = nombre de pages dans les sections de programme "examen" et "correction"

y = numéro d'une ligne à corriger

y1 = plus petit numéro de ligne dans les sections de programme "insertion" et "suppression"

y2 = plus grand numéro de ligne dans les sections de programme "insertion" et "suppression"

y3 = nombre de lignes à insérer ou nombre de lignes à supprimer

z = nombre de lignes d'une page

zl = nombre de lignes d'une page dans la section de programme "correction"

zs = lignes par page

zz = caractères par ligne

Description du programme:

Lignes 10-60 : Titre

Lignes 70-170 : Définition des caractères spéciaux français

Ligne 180 : Réservation de place mémoire pour le texte

Ligne 190 : Cette ligne de programme interdit une longue réorganisation de la mémoire pour les fichiers de texte encore à ouvrir. Cette ligne peut cependant également être négligée car le fait de lancer à nouveau le programme provoque un message d'erreur en ligne 100. Il est cependant également possible de supprimer les lignes 70 à 170 avec l'instruction DELETE avant de lancer le programme une seconde fois ou de transformer la ligne 100 en instruction REM.

Lignes 200-350 : Menu avec saut aux sous-programmes correspondants

Ligne 360 : Comme le mode écran est souvent modifié dans les sous-programmes, c'est ce qui est fait ici aussi pour garantir une sortie toujours identique du menu

Lignes 370-630 : Sous-programme "entrée du texte"
370-390 : commentaire

400-460 : affectation de valeurs aux paramètres zz et zs si cela n'a pas encore été fait et sortie d'une explication pour terminer l'entrée

470 : Saut au sous-programme "masque écran"

480 : Définition d'une zone de l'écran réservée à l'entrée de texte (la limite droite de la fenêtre dépend du paramètre "caractères par ligne")

490 : La variable s est fixée sur 1 lors de l'entrée de la toute première ligne (s était auparavant égal à 0)

500-540 : Sortie du texte d'une page partiellement écrite

550 : Si une page est pleine (alors z=zs) on passe à une autre page (s=s+1) et la variable z est fixée égale à 0

560 : Les variables z et az sont augmentées de 1 avant toute entrée de ligne

570 : Sortie d'un commentaire et retour au menu si 300 lignes de texte figurent dans la mémoire programme

580-590 : Sortie du nombre de pages et du numéro de ligne (par rapport à une page)

600 : Entrée d'une ligne de texte. L'index d'une ligne de texte est toujours le numéro de ligne absolu az.

610-630 : Les variables z et az sont diminuées de 1 lors de l'entrée du critère d'interruption (le critère d'interruption ne fait pas en effet partie du texte proprement dit), la zone écran est redéfinie et on quitte ce sous-programme. Par ailleurs on saute à la ligne 550 pour la poursuite de l'entrée du texte.

Lignes 640-830 : Sous-programme "examen du texte"

640-660 : Commentaire

670-680 : Entrée d'un nombre de pages
690-700 : La variable auxiliaire 1 est fixée à 0, on saute au sous-programme "masque écran" et la fenêtre de texte est redéfinie
710 : Sortie du nombre de pages
720 : Les numéros de ligne (absolus) des début et fin d'une page sont déterminés
730-780 : Sortie du texte. Après sortie de 22 lignes (voir variable auxiliaire 1) on saute au sous-programme "attendre". On renonce ici au message "Frappez une touche S.V.P.".
790-830 : La fenêtre de texte est redéfinie et on peut examiner d'autres pages. Par ailleurs, retour du sous-programme.

Lignes 840-1710 : Sous-programme "correction du texte"
840-860 : Commentaire
870-970 : Menu et saut correspondant ainsi que demande d'un nombre de pages
980-1130 : Section de programme "correction"
Un numéro de ligne est réclamé (lignes 1010-1020), le numéro de ligne absolu correspondant est calculé (1030), la ligne à corriger est sortie avec les lignes voisines (1040-1090) puis corrigée (1100-1120). L'écran est enfin vidé et on retourne au menu.
1140-1450 : Section de programme "insertion"
Les numéros de ligne sont réclamés, les nombres caractéristiques absolus sont calculés et les lignes de texte sont sorties (1170-1290).

Une valeur est alors entrée pour le nombre de lignes à insérer (1300-1310), les variables z et az sont rectifiées (1320-1330, en tenant compte en ligne 1330 du fait que les lignes supplémentaires nécessitent peut-être une nouvelle page), les indices des lignes de texte existantes sont rectifiés (1340-1360) et les nouvelles lignes de texte peuvent être insérées grâce à la définition de la fenêtre écran et au sous-programme "masque écran" (1360-1440). On revient ensuite au menu (1450).

1460-1690 : Section de programme "suppression"
Les numéros de ligne sont réclamés, convertis et les lignes correspondantes sont sorties (1490-1600). On calcule ensuite (1630), après une sortie de commentaire (1610) le nombre de lignes à supprimer (1640-1660). Les variables z et az sont alors à nouveau calculées (1670) en tenant compte ici aussi d'un changement éventuel du nombre de pages (1680). Enfin, retour au menu (1690).

1700-1710 : Fin des corrections: vidage de l'écran et retour du sous-programme

Lignes 1720-1810 : Sous-programme "chargement d'un texte"
Le chargement s'effectue par étapes de travail différentes selon qu'il s'agit de variables numériques ou de variables de texte. Pour le travail avec la disquette, la ligne 1750 doit être modifiée car il faut alors fournir un nom de fichier concret.

Lignes 1820-1930 : Sous-programme "sauvegarde de textes"

La sauvegarde s'effectue exactement comme le chargement. Vous pouvez choisir librement le nom du texte à sauvegarder. Il faut noter cependant qu'on travaille ici en ligne 1900 non pas avec l'instruction PRINT mais avec l'instruction WRITE, c'est-à-dire que les lignes de texte sont sauvegardées entre guillemets pour empêcher par exemple que les virgules ne conduisent à des coupures de phrase indésirables lors de la sortie.

Lignes 1940-2240 : Sous-programme "Impression du texte"

Après quelques indications de format (1970-2040), le nombre de caractères par ligne est limité (2050) et le texte est imprimé en totalité ou page par page. La ligne 2160 est identique à la ligne 720.

Dans cette section de programme peuvent encore être ajoutées des instructions supplémentaires pour la commande de l'imprimante (menu d'impression). L'instruction TAB pose notamment des problèmes pour les textes importants. Il vaut mieux alors utiliser des instructions spécifiques pour l'imprimante telles que CHR$(27);'l';CHR$(n);(imprimanteEpson), où "n" indique l'emplacement en colonnes de la sortie.

Lignes 2250-2360 : Sous-programme "fin du travail"

Si vous avez oublié de sauvegarder votre texte, vous pouvez encore le faire ici. Sinon, le programme est terminé.

Lignes 2370-2470 : Sous-programme "masque écran"
Le masque est créé en définissant deux fenêtres écran.
La première fenêtre (définie en ligne 2410) sert à la sortie des nombres de lignes et de pages actuels. La limite d'une ligne est dessinée (2450-2460) dans la seconde fenêtre (définie en ligne 2440). La définition des fenêtres de texte présente l'avantage de conserver le "masque" même si les lignes de texte disparaissent de l'écran (zone écran numéro 0).

Lignes 2480-2550 : Sous-programme "attendre"

5.5 Le puits à idées

- Voeux d'anniversaire
- Invitation à une soirée
- Lettres standard et lettres commerciales (factures, rappels, etc.).
- Candidatures (Lettre et curriculum vitae)
- Lettres publicitaires personnalisées
- Documents
- Sauvegarde et impression d'adresses
- Impression d'étiquettes informatiques autocollantes
- Présentation professionnelle d'exposés et de thèses
- et bien d'autres choses

6. Economie domestique et santé

6.1 Remarque préalable

Disons-le tout de suite: nous ne présentons pas dans ce chapitre de programme de gestion de comptabilité familiale. Un budget domestique ne se gère de toute façon pas comme une PME.

Nous allons plutôt vous donner une aide technique en programmation pour vous permettre de conserver ou d'essayer de conserver le contrôle de tous les coûts qui grèvent votre gestion domestique.

Nous traiterons en outre dans ce chapitre de la santé. Il y a de nombreuses possibilités d'utiliser votre CPC dans le domaine de la santé. Pensez par exemple aux plans de jogging ou de régime. Nous nous limiterons cependant à un programme de calculateur de calories. Ce programme est très pratique car il peut par exemple être converti sans problème pour les diabétiques en calculateur d'unités de pain. On peut bien sûr l'utiliser également pour d'autres valeurs nutritives.

6.2 Un budget domestique rationnel

Plus vous employez votre argent de façon rationnelle et plus vous pouvez vous permettre de choses. Cette opinion très répandue est certainement fondée. Mais que veut dire rationnel? Pour le consommateur, et nous sommes tous des consommateurs, il est souvent à peine possible de juger de la qualité d'un produit. D'autre part les prix bougent sans cesse dans notre système économique. Il n'est donc pas si simple d'avoir un comportement rationnel.

Il est donc peut-être judicieux de se demander pour quoi est dépensé l'argent du ménage. Si vous ne vous êtes jamais encore posé cette question, vous risquez d'être fort étonné une fois que vous découvrirez comment vos dépenses se répartissent entre les différentes catégories que sont les vêtements, l'alimentation, les voyages, etc. Un tel contrôle de vos sorties d'argent comprend bien sûr une grande masse d'informations. C'est pourquoi l'emploi d'un CPC peut être fort utile pour la gestion de ces données.

Le programme suivant se charge de cette tâche en offrant les fonctions suivantes:

- Entrée de sorties d'argent (avec les variables "nom marchandise"), "montant francs", "catégorie" et "date").

- Constitution de catégories (jusqu'à 10 catégories peuvent être constituées).

- Pour les sorties d'argent et les catégories:

- lecture

- sauvegarde

- examen (sous forme de tableau)

- calcul (les montants en francs par catégorie sont sortis pour n'importe quelle période. La représentation de la répartition en pourcentage des différentes catégories se fait sous la forme d'un histogramme.)

- Fin du travail

Le programme ne gère donc pas seulement les différentes informations mais il prend également en charge la tâche importante qui consiste à effectuer des calculs. Vous pouvez ainsi constater en fin de mois ou en fin d'année combien vous avez dépensé d'argent pour chaque catégorie de dépenses.

Comme vous ne pouvez constituer que 10 catégories (un plus grand nombre de catégories nuirait à la clarté du tableau), il est recommandé de créer une catégorie "divers" pour éviter des problèmes insolubles de classement. Vous devez effectuer une lecture (chargement à partir de la cassette ou de la disquette) AVANT d'entrer de nouvelles données car sinon les informations que vous avez entrées avant le chargement seront effacées.

Pour le travail avec la disquette, il convient de modifier la routine de lecture puisqu'il faut absolument fournir un nom de fichier. Cela peut se présenter ainsi:

```
795 INPUT"Nom du fichier ";n$
800 OPENIN n$
```

Il y a place en tout dans le programme pour un maximum de 1000 entrées. Vous pouvez bien sûr réhausser cette limite (lignes 80-90 mais attention à la capacité mémoire de votre ordinateur) mais il serait certainement plus judicieux d'établir un bilan final pour chaque mois ou chaque année.

Pour le reste, le programme est largement auto-explicatif. Nous avons renoncé après la sortie de l'histogramme au message "Frappez une touche S.V.P." car la place disponible sur l'écran est intégralement employée pour le graphique et il s'agissait d'éviter un glissement de l'écran vers le haut.

Programme:

```
10 REM I15
20 CLS
30 PRINT "I15 - Programme de controle"
40 PRINT TAB(7)"de vos depenses"
50 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
60 PRINT TAB(9) "Bernd Kowal, 1985"
70 GOSUB 2060
80 DIM wb$(1000),dm(1000),k(1000)
90 DIM tg(1000),mo(1000),ja(1000),ja$(1000)
100 REM
110 REM Menu
120 REM
130 PRINT TAB(17)"Menu":PRINT:PRINT
140 PRINT TAB(33)"Entree":PRINT
150 PRINT"Entrer depenses";TAB(36)"1":PRINT
160 PRINT"Creer categories";TAB(36)"2":PRINT
170 PRINT"Depenses et categories :":PRINT
180 PRINT" charger";TAB(36)"3":PRINT
190 PRINT" sauvegarder";TAB(36)"4":PRINT
200 PRINT" examiner";TAB(36)"5":PRINT
210 PRINT" calculer";TAB(36)"6"
220 PRINT:PRINT"Fin du travail";TAB(36)"7"
230 PRINT:PRINT:INPUT"Votre choix ";a:CLS
240 IF a<1 OR a>7 THEN CLS:GOTO 130
250 ON a GOTO 270,620,780,920,1080,1240,1930
260 REM
270 REM Entrer les depenses
280 REM
290 IF ka>0 THEN 320
300 PRINT"Vous devez d'abord creer des categories."
310 GOSUB 2050:GOTO 620
320 WINDOW #2,20,40,1,25
330 PRINT #2,"Categories":PRINT #2
340 FOR i=1 TO ka
350 PRINT #2,i;".";k$(i)
360 NEXT i
370 WINDOW #1,1,19,1,25
380 ga=ga+1
390 PRINT #1,"L'entree s'acheve"
400 PRINT #1,"si vous entrez"
410 PRINT #1,"un 0 comme nom"
420 PRINT #1,"de marchandise":PRINT#1
430 PRINT #1,"Veuillez entrer"
440 PRINT #1,"les informations"
450 PRINT #1,"suivantes:":PRINT #1:PRINT #1
460 PRINT #1,"Entree";ga;":":PRINT #1
470 PRINT #1,"Nom de marchandise"
480 INPUT #1,wb$(ga):PRINT #1
```

```
490 IF wb$(ga)="0" THEN ga=ga-1:CLS:GOTO 110
500 PRINT #1,"Montant en FF"
510 INPUT #1,dm(ga):PRINT #1
520 PRINT #1,"Categorie"
530 INPUT #1,"Nr. ";k(ga):PRINT #1
540 IF k(ga)<1 OR k(ga)>ka THEN 520
550 PRINT #1,"Date"
560 INPUT #1,"Jour   ";tg(ga)
570 INPUT #1,"Mois   ";mo(ga)
580 INPUT #1,"Annee  ";ja$(ga)
590 ja$(ga)=RIGHT$(ja$(ga),2):ja(ga)=VAL(ja$(ga))
600 CLS #1:GOTO 380
610 REM
620 REM Creer des categories
630 REM
640 IF ka=0 THEN 690
650 PRINT"Les categories suivantes existent deja:"
660 PRINT:FOR i=1 TO ka
670 PRINT "categorie";i;":";k$(i)
680 NEXT i:PRINT:PRINT
690 PRINT"Vous pouvez creer 10 categories maximum."
700 PRINT"Entrez un 0 si vous ne voulez plus"
710 PRINT"creer de nouvelle categorie.":PRINT
720 ka=ka+1
730 IF ka=11 THEN ka=ka-1:GOSUB 2050:GOTO 110
740 PRINT "Categorie";ka;:INPUT k$(ka)
750 IF k$(ka)="0" THEN ka=ka-1:CLS:GOTO 110
760 GOTO 720
770 REM
780 REM Charger depenses et categories
790 REM
800 OPENIN ""
810 INPUT #9,ga,ka
820 FOR i=1 TO ga
830 INPUT #9,wb$(i)
840 INPUT #9,dm(i),k(i),tg(i),mo(i),ja(i)
850 NEXT i
860 FOR i=1 TO ka
870 INPUT #9,k$(i)
880 NEXT i
890 CLOSEIN
900 CLS:GOTO 110
910 REM
920 REM Sauvegarder depenses et categories
930 REM
940 PRINT"Sous quel nom les informations doivent-"
950 INPUT"elles etre sauvegardees";n$
960 OPENOUT n$
```

```
970 PRINT #9,ga,ka
980 FOR i=1 TO ga
990 PRINT #9,wb$(i)
1000 PRINT #9,dm(i),k(i),tg(i),mo(i),ja(i)
1010 NEXT i
1020 FOR i=1 TO ka
1030 PRINT #9,k$(i)
1040 NEXT i
1050 CLOSEOUT
1060 CLS:GOTO 110
1070 REM
1080 REM examiner depenses et categories
1090 REM
1100 MODE 2:af=1:h=20
1110 PRINT"No d'entree   Nom de marchandise  ";
1120 PRINT"Montant FF    Categorie           Date"
1130 PRINT:FOR i=af TO ga
1140 PRINT USING "    #####    ";i;
1150 PRINT USING "   \                    \";wb$(i);
1160 PRINT USING "     ######.##";dm(i);
1170 PRINT USING "     \             \      ";k$(k(i));
1180 PRINT USING "##.";tg(i);mo(i);
1190 PRINT USING "##";ja(i)
1200 IF i=h THEN h=h+20:af=af+20:GOSUB 2060:GOTO 1110
1210 NEXT i
1220 GOSUB 2050:MODE 1:GOTO 110
1230 REM
1240 REM Calculs
1250 REM
1260 PRINT"Veuillez entrer la periode que doivent"
1270 PRINT"concerner les calculs.":PRINT
1280 PRINT"de :";TAB(18)"a :":PRINT
1290 PRINT:PRINT"Jour";TAB(18)"Jour":PRINT
1300 PRINT"Mois";TAB(18)"Mois":PRINT
1310 PRINT"Annee";TAB(18)"Annee"
1320 LOCATE 7,7:INPUT at
1330 LOCATE 7,9:INPUT am
1340 LOCATE 7,11:INPUT aja$
1350 LOCATE 24,7:INPUT et
1360 LOCATE 24,9:INPUT em
1370 LOCATE 24,11:INPUT eja$
1380 aja$=RIGHT$(aja$,2):aja=VAL(aja$)
1390 eja$=RIGHT$(eja$,2):eja=VAL(eja$)
1400 PRINT:PRINT:PRINT TAB(8)"Je calcule !"
1410 su=0:FOR i=1 TO ka
1420 ks(i)=0:NEXT i
1430 FOR i=1 TO ga
1440 IF ja(i)<aja OR ja(i)>eja THEN 1510
```

```
1450 IF ja(i)>aja AND ja(i)<eja THEN 1490
1460 IF mo(i)<am OR mo(i)>em THEN 1510
1470 IF mo(i)>am AND mo(i)<em THEN 1490
1480 IF tg(i)<at OR tg(i)>et THEN 1510
1490 ks(k(i))=ks(k(i))+dm(i)
1500 su=su+dm(i)
1510 NEXT i
1520 CLS:PRINT"Dans cette periode, vous avez depense"
1530 PRINT"au total";INT(su*100+0.5)/100;"FF."
1540 PRINT:PRINT
1550 PRINT"Pour les differentes categories, cela"
1560 PRINT"donne l'image suivante :":PRINT
1570 PRINT"No de la   Nom de la";TAB(35)"FF"
1580 PRINT"Categorie  Categorie":PRINT
1590 FOR i=1 TO ka
1600 PRINT USING "#####";i;
1610 PRINT USING "     \              \";k$(i);
1620 z=INT(ks(i)*100+0.5)/100
1630 PRINT TAB(30) USING "#######.##";z
1640 NEXT i
1650 GOSUB 2050
1660 PRINT"Representation graphique de la"
1670 PRINT"repartition en pourcentages arrondis"
1680 PRINT"des differentes categories de depense:"
1690 pm=0
1700 FOR i=1 TO ka
1710 p(i)=ks(i)*100/su
1720 IF p(i)>pm THEN pm=p(i)
1730 NEXT i
1740 FOR i=1 TO ka
1750 pp(i)=p(i)*100/pm
1760 NEXT i
1770 PLOT 17,17:DRAW 17,320
1780 PLOT 17,17:DRAW 520,17
1790 LOCATE 1,5:PRINT"%"
1800 LOCATE 34,25:PRINT"Categ."
1810 j=0
1820 FOR i=3 TO ka*3 STEP 3
1830 j=j+1
1840 LOCATE i,25:PRINT j
1850 x=ROUND(18*pp(j)/100)
1860 FOR l=24 TO 24-x STEP -1
1870 LOCATE i+1,l:PRINT CHR$(143)
1880 NEXT l
1890 LOCATE i,23-x:PRINT USING "##"; p(j)
1900 NEXT i
1910 GOSUB 2100:GOTO 110
1920 REM
```

```
1930 REM Fin du Programme
1940 REM
1950 PRINT"Etes-vous sur de bien avoir sauvegarde"
1960 PRINT"les informations comme il convient ?"
1970 PRINT"Si ce n'est pas le cas, entrez"
1980 PRINT"simplement un 'oh' (sinon, la touche"
1990 PRINT"espace suffit)."
2000 INPUT a$
2010 IF a$="oh" THEN CLS:GOTO 920
2020 PRINT:PRINT:PRINT"         Au revoir"
2030 PRINT:PRINT"Vous pouvez maintenant me debrancher"
2040 END
2050 REM
2060 REM SP Attendre
2070 REM
2080 LOCATE 5,25
2090 PRINT"Frappez une touche S.V.P"
2100 x$=INKEY$:IF x$="" THEN 2100
2110 CLS:RETURN
```

Liste de variables:

a = réponse au menu

a$ = chaîne de réponse dans la section de programme "fin du travail"

af = valeur initiale de la boucle

aja = indication de l'année (début)

aja$ = tableau d'entrée pour aja

am = indication du mois (début)

at = indication du jour (début)

dm (i) = montant en francs

eja = indication de l'année (fin)

eja$ = tableau d'entrée pour eja

em = indication du mois (fin)

et = indication du jour (fin)

ga = nombre de sorties d'argent

h = variable auxiliaire

i = index de comptage

j = index

ja(i) = année de la sortie d'argent

ja$(i) = tableau d'entrée pour ja(i)

k(i) = numéro de catégorie pour une sortie d'argent

k$(i) = désignation de la catégorie

ka = nombre de catégories définies

ks(i) = somme des sorties d'argent pour une catégorie donnée dans une période donnée

l = valeur initiale de la boucle de dessin de l'histogramme

mo(i) = mois de la sortie d'argent

n$ = nom des informations à sauvegarder

p(i) = pourcentage des sorties d'argent par catégorie par rapport à l'ensemble des sorties d'argent

pm = valeur maximale par rapport à la variable p(i)

pp(i) = pourcentage des sorties d'argent par catégorie par rapport à la valeur maximale pm

su = somme des sorties d'argent pour une période donnée

tg(i) = jour de la sortie d'argent

wb$(i) = désignation de la marchandise

x = variable auxiliaire pour déterminer la valeur finale de la boucle de dessin des histogrammes

z = valeur arrondie à deux décimales après la virgule pour la variable ks

Description du programme:

Lignes 10-70 : Titre

Lignes 80-90 : Réservation de place mémoire

Lignes 100-250 : Menu et branchement en conséquence

Lignes 260-600 : Entrée des sorties d'argent

Dans cette section, deux fenêtres écran sont définies. Alors que les entrées se font dans la première fenêtre (zone gauche de l'écran, définie en ligne 370), les catégories existantes sont sorties dans la seconde fenêtre (zone droite de l'écran, définie en ligne 320). L'utilisateur du programme a ainsi toujours sous les yeux les différentes catégories existantes. Une fois l'entrée terminée, on ne vide en ligne 600 que la première fenêtre. Si aucune catégorie n'a encore été définie (voir lignes 290-310) un commentaire est sorti et on saute à la section de programme "Constitution de catégories". L'entrée d'un 0 pour la variable "désignation de la marchandise" a pour effet un retour au menu (ligne 490). Le tableau d'entrée pour la variable ja est une chaîne ce qui permet en ligne 590 de ne retenir dans le tableau ja(i) que les deux derniers chiffres de l'année (par exemple 85) si vous entrez le nom de l'année avec le siècle (par exemple 1985).

Lignes 610-760 : Constitution des catégories

Si des catégories ont déjà été définies, celles-ci sont sorties (lignes 640-680). On peut alors définir des catégories. Une fois que 10 catégories ont été créées ou qu'un 0 a été entré, on retourne au menu (voir lignes 730 et 750).

Lignes 770-900 : Lecture des sorties d'argent et catégories sauvegardées

Lignes 910-1060 : Sauvegarde des sorties d'argent et catégories

Lors de la lecture ou de la sauvegarde de variables alphanumériques, il faut noter que cela se produit chaque fois avec une instruction INPUT ou PRINT particulière (voir lignes 830-840 ou lignes 990-1000).

Lignes 1070-1220 : Examen des sorties d'argent et catégories

Le mode écran est fixé à cet effet sur 80 colonnes en ligne 1100. Une sortie formatée sous forme de tableau est alors réalisée à l'aide de l'instruction PRINT USING. Les variables af et h servent à limiter la sortie à 20 lignes de tableau à la fois. Avant le retour au menu, on revient en mode 40 colonnes (ligne 1220).

Lignes 1230-1910 : Calculs

1260-1390 : Entrée de la période que doivent concerner les calculs. Le texte d'entrée est d'abord sorti, puis les instructions INPUT sont positionnées en conséquence. Les indications d'année sont entrées comme chaînes pour les mêmes raisons qu'en ligne 590.

1400 : Commentaire

1410-1420 : Les variables à calculer su et ks(i) sont fixées sur la valeur initiale 0.

1430-1510 : Si une sortie d'argent se situe dans la période indiquée, les variables su et ks(i) sont augmentées du montant en francs correspondant. La comparaison des indications d'année, mois et jour pour une sortie d'argent avec la période indiquée ne fonctionne pour ainsi dire que pour notre siècle puisque le siècle n'est pas pris en compte dans l'indication de l'année. L'année 2005 sera par exemple traitée exactement comme l'année 1905.

1520-1650 : Sortie formatée des montants en francs (par catégorie et montant global) arrondis à deux décimales.

1660-1680 : Titre pour la représentation graphique de la répartition en pourcentage des différentes catégories de sorties d'argent.

1690 : La variable pm est fixée à 0.

1700-1730 : Calcul des pourcentages des sorties d'argent par catégorie par rapport au total des sorties et détermination du pourcentage maximum pm.

1740-1760 : Calcul des pourcentages des sorties par catégorie par rapport à la valeur maximale de pourcentage pm. Cela sert à optimiser l'utilisation de la place disponible sur l'écran (voir lignes 1850-1880).

1770-1780 : Dessin d'axes de coordonnées. Avant que les lignes ne soient dessinées avec l'instruction DRAW, le curseur graphique est chaque fois placé sur l'origine des axes avec l'instruction PLOT.

1790-1800 : Ecriture des axes de coordonnées
1810 : La variable d'index j est fixée sur 0.
1820-1900 : Ecriture de l'axe des x dans cette boucle et dessin des histogrammes. La variable i sert au positionnement horizontal du curseur alors que la variable j identifie les valeurs en pourcentage pp(i) des différentes catégories. La variable x atteint son maximum avec la plus grande valeur en pourcentage, de sorte que les 18 lignes disponibles pour le dessin des histogrammes (lignes 1860-1880) sont utilisées. Pour des valeurs pp(i) plus petites, la valeur finale de la boucle est réduite en proportion (ligne 1860) de sorte que des histogrammes plus courts sont alors dessinés. En ligne 1890, les pourcentages de chaque catégorie sont sortis directement au dessus de l'histogramme correspondant. Comme il s'agit là de valeur arrondies, il se peut que la somme des pourcentages indiqués ne soit pas égale à 100.
1910 : Fin de cette section de programme

Lignes 1920-2040 : Fin du programme

On peut encore, dans cette section de programme, sauvegarder les données entrées si on a oublié de le faire.

Lignes 2050-2120 : Sous-programme "attendre"

6.3 Un calculateur électronique de calories

Tous ceux qui ont déjà fait un régime savent bien qu'il n'est pas si facile de calculer sa ration quotidienne de calories. Les indications fournies dans les tableaux de calories se rapportent en effet en général à une quantité déterminée en grammes ou en litres.

Vous vous trouvez par exemple devant le problème de règle de trois suivant:

> une bouteille de bière (de 0.5 litres) contient 210 kilocalories, combien de kilocalories x contiendra un verre (0.2 litres) ?

Vous devez poser une équation simple pour résoudre ce problème de calcul:

$$\frac{210}{0.5} = \frac{x}{0.2}$$

$$x = 0.2 * 210/0.5 = 84$$

Si vous voulez effectuer ce calcul pour chaque aliment que vous prenez quotidiennement, vous aurez beaucoup de travail. Il serait au minimum conseillé de se munir d'une calculatrice de poche. Mais vous pouvez encore gagner beaucoup plus de temps pour ces calculs si vous utilisez votre CPC avec un programme approprié.

Vous n'aurez plus en effet à faire le travail qu'une fois pour toutes en entrant dans le programme tous les aliments que vous prenez régulièrement.

L'entrée se fait avec l'instruction DATA. Chaque instruction DATA doit avoir la structure suivante:

DATA <aliment, unité de mesure, quantité d'unités de mesure à laquelle se rapporte l'indication des calories, indication des calories>

par exemple:

DATA <petit pain, pièces (de 40 grammes), 1, 111>

en d'autres termes: 1 petit pain (de 40 grammes) contient 111 kilocalories.

Dans le programme que nous vous présentons, nous avons écrit 15 lignes DATA à titre d'exemple. Vous pouvez augmenter ou modifier à votre guise cette partie du programme. Il vous faut, le cas échéant, modifier également la valeur de la variable k en ligne 100 car seuls k instructions DATA seront lues.
Lorsque vous utilisez le programme, vous n'avez à entrer pour le calcul de calories que l'aliment et un nombre se rapportant à la quantité, par exemple:

Aliment? Petit pain
Combien de pièces (de 40 grammes) ? 2.5

Vous obtenez alors comme résultat:

Cela fait 277.5 kilocalories.

Les différents résultats sont naturellement additionnés de sorte que vous pourriez avoir pour la ration alimentaire d'un jour:

Cela fait au total 2428.5 kilocalories.

Faites attention lors de l'entrée de l'aliment à bien respecter l'orthographe employée en DATA, sinon le message:

Il n'y a pas d'informations
concernant l'aliment indiqué

apparaîtra à l'écran.

Programme:

```
10 REM I16
20 CLS
30 PRINT "I16 - Programme de calcul"
40 PRINT TAB(7)"de calories"
50 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
60 PRINT TAB(9) "Bernd Kowal, 1985"
70 GOSUB 1480
80 REM k correspond au nombre d'aliments
90 REM entres
100 k=15
110 DIM n$(k),m$(k),am(k),ka(k)
120 DATA beurre,grammes,100,775
130 DATA oeuf,pieces (calibre 4),1,92
140 DATA banane,pieces (petite),1,99
150 DATA petit pain,pieces (soit 40 grammes),1,111
160 DATA anguille,grammes,100,299
170 DATA frites,grammes,100,538
180 DATA Cantal (40%),grammes,100,331
190 DATA biere ambree,litres,0.2,84
200 DATA lait (3.5%),litres,1,660
210 DATA pates (cuites),cuilleres (soit  40 grammes),1,3
9
220 DATA biere forte,litres,0.2,132
230 DATA cuissot de chevreuil,grammes,100,106
240 DATA ail, gousse (5 grammes),1,7
250 DATA confiture,petites cuilleres (soit 5 grammes),1,
38
260 DATA choux de Bruxelles,grammes,100,52
1000 FOR i=1 TO k
1010 READ n$(i),m$(i),am(i),ka(i)
1020 NEXT i
1030 PRINT TAB(16)"Menu":PRINT:PRINT:PRINT
1040 PRINT TAB(33)"Entree":PRINT:PRINT
1050 PRINT"examiner toutes les donnees";TAB(36)"1":PRINT
1060 PRINT"calculer les calories";TAB(36)"2":PRINT
1070 PRINT"Fin du travail";TAB(36)"3":PRINT
1080 PRINT:PRINT:INPUT"Votre choix ";a
1090 IF a<1 OR a>3 THEN CLS:GOTO 1030
1100 CLS:ON a GOTO 1110,1260,1470
1110 FOR i=1 TO k-1
1120 FOR j=i+1 TO k
1130 IF n$(i)<=n$(j) THEN 1180
1140 h$=n$(j):n$(j)=n$(i):n$(i)=h$
1150 h$=m$(j):m$(j)=m$(i):m$(i)=h$
1160 h=am(j):am(j)=am(i):am(i)=h
1170 h=ka(j):ka(j)=ka(i):ka(i)=h
1180 NEXT j
```

```
1190 NEXT i
1200 l=20
1210 FOR i=1 TO k
1220 PRINT n$(i)
1230 IF i=l THEN l=l+20:GOSUB 1480
1240 NEXT i
1250 GOSUB 1480:GOTO 1030
1260 ksu=0
1270 PRINT"Si vous actionnez simplement la touche"
1280 PRINT"espace en guise d'entree d'un"
1290 PRINT"aliment, le calcul des calories"
1300 PRINT"sera interrompu.":PRINT:PRINT
1310 INPUT"Aliment ";en$:PRINT
1320 IF en$="" THEN 1390
1330 FOR i=1 TO k
1340 IF n$(i)<>en$ THEN 1430
1350 PRINT"Combien de ";m$(i);" ";:INPUT em
1360 ks=em*ka(i)/am(i):PRINT:PRINT
1370 PRINT"Cela represente";ks;"kilocalories.":PRINT
1380 ksu=ksu+ks
1390 PRINT:PRINT"Ce qui donne au total";
1400 PRINT ksu;"kilocalories."
1410 GOSUB 1480
1420 IF en$="" THEN 1030 ELSE 1270
1430 NEXT i
1440 PRINT:PRINT"Il n'y a pas d'informations"
1450 PRINT"concernant l'aliment indique."
1460 GOSUB 1480:GOTO 1270
1470 END
1480 REM SP attendre
1490 LOCATE 5,25
1500 PRINT"Frappez une touche S.V.P !"
1510 x$=INKEY$
1520 IF x$="" THEN 1510
1530 CLS:RETURN
```

Liste de variables:

a = réponse au menu

am(i) = nombre d'unités auquel se refère l'indication des calories

em = nombre d'unités de mesure entrées

en$ = aliment entré

i = index de comptage

j = index de comptage

k = nombre d'instructions DATA

ka(i) = indication des calories

ks = nombre de calories calculé

ksu = somme des calories calculées

l = variable auxiliaire

m$(i) = unité de mesure

n$(i) = aliment

Description du programme:

Lignes 10-70 : Titre

Lignes 80-90 : Commentaire

Ligne 100 : Définition de la variable k

Ligne 110 : Réservation de place en mémoire

Lignes 120-260 : Fourniture de données au programme

Lignes 1000-1020 :Lecture des données

Lignes 1030-1100 :Menu avec branchement correspondant

Lignes 1110-1250 :Section de programme "Examen des données"
- 1110-1190 :Tri alphabétique des données (voir également à ce sujet le chapitre 2.4)
- 1200-1240 :Sortie chaque fois de 20 (voir variable auxiliaire l) aliments
- 1250 : "attendre" et retour au menu

Lignes 1260-1460 :Section de programme "calcul des calories"
- 1260 : la variable ksu est fixée sur zéro
- 1270-1300 :sortie d'explications sur le déroulement du programme
- 1310 : entrée d'un aliment
- 1320 : si la touche espace est enfoncée, un saut est effectué à la sortie du résultat global

1330-1430 :la variable en$, définie par une entrée (ligne 1310) est comparée aux variables stockées n$(i). En cas d'égalité, une valeur concernant la quantité est réclamée (1350), le nombre de calories est calculé (1360), le résultat est sorti (1370) et la somme de tous les calculs est formée dans le tableau ksu (1380). En lignes 1390 et 1400 est alors sorti le résultat provisoire (ou le résultat global, voir ligne 1320). Après un saut au sous-programme "attendre" (1410), a lieu en ligne 1420 un saut au menu (si les calculs doivent être interrompus) ou à une nouvelle entrée (1270). En cas de non-égalité, la comparaison se poursuit (saut à la ligne 1430).

1440-1460 :Sortie d'un commentaire, "attendre" et saut à une nouvelle entrée si le nom de l'aliment entré en$ ne fait pas partie des variables stockées sous n$(i).

Ligne 1470 : Fin du programme

Lignes 1480-1530 :Sous-programme "attendre"

6.4 Le puits à idées

- listes de prix à consulter, modifiables, par exemple pour les légumes et les fruits.

- organisation de l'économie ménagère

- rangement du congélateur

- fichier de recettes de cuisine

- archives sanitaires personnelles (consulter, calculer et évaluer valeurs de sucre dans le sang, circulation, pouls etc.).

- plan de régime

- plans de jogging et d'entretien de la forme

- calcul de risques (par exemple espérance de vie en fonction des quantités de tabac fumées).

- et bien d'autres choses

7. Votre ordinateur et vos hobbys

7.1 Remarque préalable

Ce chapitre est peut-être l'un des plus intéressants de cet ouvrage puisqu'il vous permet d'utiliser le CPC pour de nombreux hobbys. Peu importe que vous collectionniez les timbres ou que vous soyez supporter d'un sport quelconque, avec l'aide d'un ordinateur et de programmes adaptés, beaucoup de choses sont plus simples à réaliser. Cela va de la prise en charge de travaux d'organisation (où trouver tel ou tel timbre) jusqu'à l'exercice direct de votre hobby (nous ne pensons pas ici à la programmation en elle-même, mais par exemple au fait d'utiliser un ordinateur pour faire de la musique).

Il n'est évidemment pas possible de décrire ici en détail toutes les applications possibles de votre ordinateur. En fait, nous avons fait une large place à deux programmes d'exemple qui doivent vous montrer les possibilités techniques de votre CPC. Il y a d'une part les excellentes caractéristiques graphiques et d'autre part les remarquables possibilités de la programmation musicale. Ces deux domaines d'application ont d'ailleurs été quelque peu négligés jusqu'ici. Un troisième exemple concerne le roi Football qui reste un des sports les plus populaires et donc le hobby de beaucoup de nos compatriotes.

De nombreux autres domaines d'application sont cependant concevables. Dans le puits à idées de ce chapitre, vous trouverez un certain nombre d'idées. Nous vous invitons donc plus que jamais à développer, en partant des propositions qui vous sont faites ici, des programmes adaptés à vos besoins personnels. La passion que vous avez pour un hobby vous aidera d'ailleurs peut-être à ne pas vous laisser décourager par les difficultés qui peuvent apparaître dans certains cas lors de la réalisation de vos projets de logiciels. Peut-être même la programmation deviendra-t-elle un de vos nouveau hobbys (si ce n'est pas déjà le cas).

7.2 Peinture, dessin et le graphisme

Dans le marché actuel de l'informatique, une tendance très nette se dessine qui va dans le sens de systèmes dotés de possibilités graphiques. Comme le clavier de l'ordinateur apparaît souvent dépassé pour réaliser du graphisme, de nouvelles interfaces pour l'interaction homme-ordinateur ont été développées ces dernières années. Il s'agit de nouvelles machines d'entrée, telles que le joystick, déjà connu de longue date, la souris ou les tableurs graphiques. Les tableurs graphiques ressemblent d'ailleurs beaucoup au papier, support conventionnel de la pensée. Nous n'évoquerons cependant pas ces machines supplémentaires d'entrée dans les développements suivants car la description du traitement graphique professionnel dépasserait certainement le cadre de cet ouvrage. Si vous disposez par contre de cet instrument simple en comparaison que constitue le joystick, vous pouvez aisément adapter le programme graphique qui vous est ici proposé (voyez le manuel d'utilisation). Le programme proposé vous permet de réaliser n'importe quelles images (graphiques ou dessins), de les sauvegarder et de les charger ensuite à nouveau dans la machine. Qu'il s'agisse d'un travail créatif ou des plans d'une maison, cela ne dépend que de vous. En tout cas les coordonnées x et y actuelles vous seront chaque fois montrées à l'écran (en haut à droite, la coordonnée y et en bas à gauche la coordonnée x), de façon à rendre possible la réalisation de dessins techniques précis.

Pour diriger votre "crayon" sur l'écran, certaines touches ont été dotées de fonctions spéciales. Lorsqu'il s'agit d'instructions INPUT, l'entrée se fait sans couleur, c'est-à-dire que le "crayon" et le "papier" ont la même couleur pour ne pas altérer l'image graphique. En général toutefois, la valeur de la touche appuyée est restituée à l'aide de l'instruction INKEY$. Il n'est pas alors nécessaire, comme vous le savez, d'actionner la touche ENTER.

Au contraire des programmes précédents, aucune explication n'est donc sortie sur le déroulement du programme. Vous devrez donc avoir en tête la signification des touches de commande ou bien utiliser la liste suivante:

Touche de commande	Signification
curseur haut	une ligne vers le haut
–>	une ligne vers la droite
curseur bas	une ligne vers le bas
<–	une ligne vers la gauche
a	ligne sans couleur ou début "effaçage"
z	ligne sans couleur ou fin "effaçage"
k	cercle

valeur rayon en pixel (gr. ENTER)
valeur angle en degrés (gr. ENTER)

d	diagonale ou ligne tracée

pour la pente:
valeur coordonnée x (horizontale) (gr. ENTER)
valeur coordonnée y (verticale) (gr. ENTER)
touche COPY tracer ligne
petite ENTER fin de cette fonction

g cercle plein
valeur rayon en pixel (gr. ENTER)
valeur angle en dégrés (gr. ENTER)

c changement de couleur
b (gr. ENTER) couleur devient bleue
r (gr. ENTER) couleur devient rouge
n (gr. ENTER) couleur devient noire

s fin du programme

Après que vous ayez appuyé sur la touche "s", entrez PEN 1 pour que le curseur redevienne visible.

Les explications sorties par le programme sont conçues pour l'utilisation avec un lecteur de cassette. Pour le travail sur disquette, il faut donc modifier en conséquence les lignes 150, 190, 260, 310 et 320. Il faut en outre choisir en lignes 200 et 300 un nom de fichier plus court.

Comme le maniement de ce programme peut présenter quelques difficultés, vous trouverez à la suite de la description du programme un petit exemple dans lequel chaque touche de commande est présentée au moins une fois. La description du programme est également plus détaillée, tel ou tel lecteur pouvant avoir du mal à bien maîtriser les instructions graphiques.

Programme:

```
10 REM I17
20 CLS
30 PRINT"I17 - Programme de creation d'un"
40 PRINT TAB(7)"dessin ou graphique, devant"
50 PRINT TAB(7)"etre sauve de facon externe"
60 PRINT TAB(7)"et pouvant etre recharge"
70 PRINT:PRINT:PRINT
80 PRINT TAB(11)"Bernd Kowal, 1985"
90 GOSUB 1100
100 BORDER 26
110 INK 0,26
120 INK 1,0
130 INK 2,6
140 INK 3,11
150 PRINT"Faut-il charger un dessin"
160 INPUT"sur cassette (o/n) ";nj$
170 IF nj$="n" THEN PRINT:PRINT:GOTO 220
180 IF nj$<>"o" THEN 150
190 PRINT:PRINT:PRINT "Press PLAY then any key :"
200 OPENIN "!graphisme"
210 INPUT #9,x,y:GOTO 340
220 PRINT"Entrez les coordonnees de depart"
230 PRINT"pour le curseur graphique :":PRINT
240 INPUT"Coordonnee x ";x
250 INPUT"Coordonnee y ";y:PRINT:PRINT
260 PRINT"Faut-il sauvegarder le graphisme sur"
270 INPUT"cassette (o/n) ";jn$
280 IF jn$="n" THEN 340
290 IF jn$<>"o" THEN 260
300 OPENOUT "!Graphisme":PRINT:PRINT
310 PRINT "Pour la sauvegarde sur cassette"
320 PRINT "Press REC and PLAY then any key :"
330 PRINT #9,x,y
340 GOSUB 1100
350 KEY DEF 0,1,72
360 KEY DEF 1,1,68
370 KEY DEF 2,1,66
380 KEY DEF 8,1,71
390 KEY DEF 9,1,100
400 KEY DEF 6,1,83
410 PLOT x,y
420 f=1
430 IF EOF=-1 AND nj$="o" THEN CLOSEIN:PEN 0:END
440 IF nj$="j" THEN INPUT #9,a$:GOTO 460
450 a$=INKEY$:IF a$="" THEN 450
460 IF a$="H" THEN y=y+1:DRAW x,y,f
470 IF a$="D" THEN x=x+1:DRAW x,y,f
480 IF a$="B" THEN y=y-1:DRAW x,y,f
```

```
490 IF a$="G" THEN x=x-1:DRAW x,y,f
500 IF a$="a" THEN f=0
510 IF a$="z" THEN f=1
520 IF a$="k" THEN 610
530 IF a$="d" THEN 730
540 IF a$="g" THEN 880
550 IF a$="c" THEN 1010
560 IF a$="S" THEN PEN 0:CLOSEOUT:END
570 LOCATE 35,25:PEN 1:PRINT x
580 LOCATE 35,1:PEN 1:PRINT y
590 IF jn$="o" THEN PRINT #9,a$
600 GOTO 430
610 REM Circonference
620 IF nj$="n" THEN 640
630 INPUT #9,r,w:GOTO 670
640 PEN 0:LOCATE 1,1:INPUT r
650 LOCATE 1,1:INPUT w:PEN f
660 IF jn$="o" THEN PRINT #9,a$:PRINT #9,r,w
670 DEG
680 FOR i=1 TO w
690 PLOT x+r*COS(i),y+r*SIN(i),f
700 NEXT i
710 PLOT x,y,f
720 GOTO 430
730 REM Diagonale
740 IF nj$="n" THEN 760
750 INPUT #9,x1,y1:GOTO 790
760 PEN 0:LOCATE 1,1:INPUT x1
770 LOCATE 1,1:INPUT y1:PEN f
780 IF jn$="o" THEN PRINT #9,a$:PRINT #9,x1,y1
790 IF EOF=-1 AND nj$="o" THEN CLOSEIN:PEN 0:END
800 IF nj$="o" THEN INPUT #9,b$:GOTO 820
810 b$=INKEY$:IF b$="" THEN 810
820 IF b$="d" THEN x=x+x1:y=y+y1:DRAW x,y,f
830 LOCATE 35,25:PEN 1:PRINT x
840 LOCATE 35,1:PEN 1:PRINT y
850 IF jn$="o" THEN PRINT #9,b$
860 IF b$="S" THEN 430
870 GOTO 790
880 REM Cercle Plein
890 IF nj$="n" THEN 910
900 INPUT #9,r,w:GOTO 940
910 PEN 0:LOCATE 1,1:INPUT r
920 LOCATE 1,1:INPUT w:PEN f
930 IF jn$="o" THEN PRINT #9,a$:PRINT #9,r,w
940 DEG
950 FOR i=1 TO w
960 PLOT x,y,f
```

```
970 DRAW x+r*COS(i),y+r*SIN(i),f
980 NEXT i
990 PLOT x,y,f
1000 GOTO 430
1010 REM Changement de couleur
1020 IF nj$="n" THEN 1040
1030 INPUT #9,fa$:GOTO 1050
1040 PEN 0:LOCATE 1,1:INPUT fa$:PEN 1
1050 IF fa$="b" THEN f=3
1060 IF fa$="r" THEN f=2
1070 IF fa$="n" THEN f=1
1080 IF jn$="j" THEN PRINT #9,a$:PRINT #9,fa$
1090 GOTO 430 .
1100 REM SP attendre
1110 LOCATE 7,25
1120 PRINT"Frappez une touche S.V.P."
1130 x$=INKEY$:IF x$="" THEN 1130
1140 CLS:RETURN
```

Liste de variables:

a$ = chaîne de caractères pour commander les déplacements sur le dessin

b$ = chaîne de caractères pour le dessin des diagonales et des lignes

f = couleur

fa$ = chaîne de caractères pour modifier la couleur

i = index de comptage

jn$ = chaîne de caractères de réponse à la question de savoir si un dessin doit être ou non sauvegardé

nj$ = chaîne de caractères de réponse à la question de savoir si un dessin doit être ou non chargé

r = rayon du cercle

w = angle

x,y = coordonnées de la "table de dessin"

x1,y1 = coordonnées de dessin d'une diagonale ou d'une ligne

Description du programme:

Lignes 10-90 : Titre

Ligne 100 : Représentation du bord de l'écran en blanc brillant

Lignes 110-140 : Les numéros 0 à 3 de PEN ou PAPER se voient affecter les couleurs blanc brillant (ligne 110), noir (ligne 120), rouge clair (ligne 130) et bleu ciel (ligne 140)

Lignes 150-160 : Demander si un dessin doit être chargé

Ligne 170 : Sortie de deux lignes vides et saut à la ligne 220 si "n" a été répondu

Ligne 180 : S'il n'a pas été répondu "o", saut à la ligne 150

Ligne 190 : Cette ligne est atteinte s'il a été répondu "o". Deux lignes vides sont sorties et une instruction pour l'utilisation du lecteur de cassette est sortie (comme pour les programmes précédents, il est ici aussi possible d'utiliser à la place un lecteur de disquette).

Ligne 200 : Ouverture du fichier d'entrée "!graphisme"

Ligne 210 : Les coordonnées de départ x et y pour le curseur graphique sont lues sur la cassette et on saute à la ligne 340

Lignes 220-250 : Les valeurs des coordonnées x et y sont réclamées

Lignes 260-290 : On demande si le dessin devant encore être réalisé doit être sauvegardé. Si "n" est entré, on saute à la ligne 340. Si "o" est entré, le programme se poursuit à la ligne 300.

Lignes 300-330 : Le fichier de sortie "!graphisme" est ouvert, un message correspondant est sorti et les valeurs des variables x et y sont stockées sur la cassette.

Ligne 340 : Saut au sous-programme "attendre"

Lignes 350-400 : Pour faciliter le maniement du programme, les touches suivantes reçoivent d'autres affectations: la touche curseur haut (ligne 350) reçoit la valeur H (comme haut), la touche -> (ligne 360) la valeur D (comme droite), la touche curseur bas (ligne 370) la valeur B (comme bas), la touche <- (ligne 380) la valeur G (comme gauche), la touche COPY (ligne 390) la valeur d (comme draw) et la petite touche ENTER (ligne 400) la valeur S (comme stop).

Ligne 410 : Dessiner le point de départ du curseur graphique

Ligne 420 : La variable f est fixée sur 1, c'est-à-dire que la couleur du crayon (ou plus exactement du curseur graphique) est sur noir jusqu'à nouvel ordre

Ligne 430 : Si nj$ égale "o" (c'est-à-dire si un dessin est chargé) et si EOF = -1 (EOF ne vaut - 1 que lorsque la fin d'un fichier d'entrée est atteinte), le fichier d'entrée est fermé et la fin du programme est atteinte. L'instruction PEN 0 a pour effet de faire apparaître pour ainsi dire sans couleur (la même couleur est utilisée pour PEN et PAPER) le message READY qui apparaît toujours normalement à l'écran à la fin d'un programme. L'image du dessin terminé ne sera ainsi pas endommagée.

Ligne 440 : Si nj$ vaut "o" mais que EOF <>-1, une nouvelle valeur est lue pour la variable a$ et on saute à la ligne 460.

Ligne 450 : Si on ne charge pas un dessin, les valeurs de la variable de commande a$ doivent être entrées en frappant une touche.

Lignes 460-490 : Suivant le caractère entré pour a$, une ligne est tracée dans la couleur actuelle (variable f), de la position actuelle du curseur à la position x, y indiquée.

Lignes 500-510 : Si a$ est égal à a (marque de début de l'effaçage), le curseur graphique reçoit la même couleur que le fond, c'est-à-dire que des lignes existantes peuvent être "effacées". Cela se produit tant que a$ est différent de z (marque de fin de l'effaçage).

Ligne 520 : Si a$ vaut k, un saut à la section de programme "dimension du cercle" (ligne 610) est effectué. Un cercle non rempli y sera alors dessiné.

Lignes 530-550 : Des sauts de programme adéquats sont effectués suivant que a$ vaut g ("cercle rempli"), d ("diagonale") ou c ("modification de couleur").

Ligne 560 : La fin du programme est atteinte (si on ne charge pas de données) si un s est entré pour a$. Le message est sorti sans couleur (voir ligne 430). L'instruction CLOSEOUT ferme le fichier de sortie. Si le dessin ne devait pas être sauvegardé, cette instruction alors inutile ne nuit pas au bon déroulement du programme.

Lignes 570-580 : Pendant le dessin, la valeur x actuelle du curseur graphique est affichée dans l'angle inférieur gauche et sa valeur y actuelle dans l'angle supérieur droit.

Ligne 590 : Le caractère pour a$ est écrit sur la cassette si le dessin doit être sauvegardé

Ligne 600 : Pour un parcours supplémentaire de la section de programme des lignes 430-590, un saut approprié est effectué

Lignes 610-720 : Dessin d'un cercle

- 610 : commentaire
- 620-630 : Si nj$ vaut o, les valeurs pour r (rayon) et w (angle) sont lues sur la cassette et on saute à la ligne 670
- 640-650 : Entrée sans couleur (pen 0) des valeurs pour r et w dans l'angle supérieur gauche de l'écran
- 660 : Si cela a été souhaité, les valeurs pour r et w sont sauvegardées. Le caractère pour a$ est également sauvegardé ici, puisqu'en ligne 720 un saut est effectué à la ligne 430 (la ligne 590 n'a pas été atteinte au cours du déroulement du programme du fait du saut en ligne 520).
- 670 : Commuter sur mesure d'angle
- 680-700 : Un cercle est dessiné en fonction de r et w: les coordonnées x,y de chaque point sont plottées dans le cercle
- 710 : Le curseur graphique est à nouveau fixé sur le centre du cercle
- 720 : Saut à la ligne 430 pour entrer un nouveau caractère de commande

Lignes 730-870 : Dessin d'une ligne ou d'une diagonale

- 730-780 : Commentaire, interrogation et entrées comme pour les lignes 610-660
- 790 : Interrogation d'EOF (voir ligne 430)
- 800-810 : Les caractères pour b$ sont chargés ou entrés (en actionnant la touche COPY)
- 820 : Suivant les valeurs de x1 et y1, une ligne est dessinée tant que b$ vaut T
- 830-840 : Affichage des coordonnées du curseur graphique

850 : Les caractères pour b$ sont sauvegardés si nécessaire

860-870 : Si la petite touche ENTER a été appuyée (ou si un "S" a été entré pour b$), un saut à la ligne 430 est effectué, sinon à l'interrogation d'EOF en ligne 790

Lignes 880-1000 : En partant chaque fois du centre (ligne 960), une ligne est dessinée vers chaque point de la superficie du cercle (ligne 970) pour représenter un cercle plein. Pour le reste, cette section de programme est identique aux lignes 610-720.

Lignes 1010-1090 :Section de programme "changement de couleur"

1010-1040 :Entrée ou chargement du caractère pour a$

1050-1070 :En entrant (ou chargeant) "b", le crayon d'écriture devient bleu, rouge pour "r" et noir pour "n"

1080-1090 :Ecrire (si jn$ vaut "o") le caractère pour fa$ sur la cassette et sauter à la ligne 430

Lignes 1100-1140 :Sous-programme "attendre"

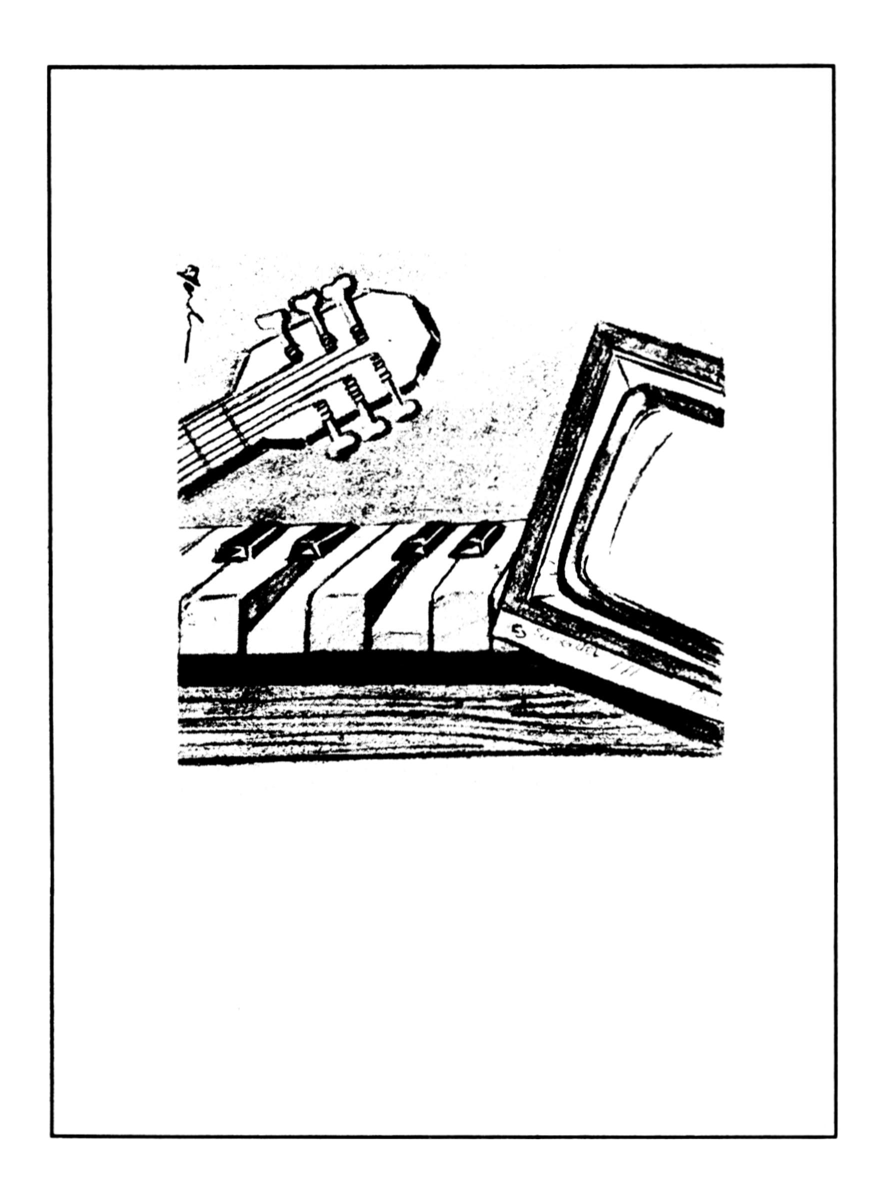
Exemple d'utilisation du programme:

Une fois le programme lancé, les coordonnées de départ se voient affecter les valeurs 120 (pour x) et 200 (pour y). Un dessin peut alors être réalisé:

Entrée au clavier	Explication
—> (jusqu'à ce que x = 170)	Une ligne vers la droite
a	Début dessin sans couleur
d	Diagonale
150 (ENTER)	150 unités vers la droite
200 (ENTER)	200 unités vers le haut
1 fois COPY	dessiner
petite ENTER	Fin de la diagonale
z	Fin dessin sans couleur
curseur bas (jusqu'à ce que y = 350)	Une ligne vers le bas
a	Début dessin sans couleur
d	Diagonale
200 (ENTER)	200 unités vers la droite
-150 (ENTER)	-150 unités vers le bas
1 fois COPY	dessiner
petite ENTER	Fin de la diagonale
z	Fin dessin sans couleur
<— (jusqu'à ce que x = 470)	Une ligne vers la gauche

a	Début dessin sans couleur
d	Diagonale
-150 (ENTER)	150 unités vers la gauche
-200 (ENTER)	200 unités vers le bas
1 fois COPY	dessiner
petite ENTER	Fin de la diagonale
z	Fin dessin sans couleur
curseur haut (jusqu'à ce que y = 50)	Une ligne vers le bas
a	Début dessin sans couleur
curseur haut (jusqu'à ce que y = 200)	Une ligne vers le bas
z	Fin dessin sans couleur
c	Changement de couleur
r (ENTER)	rouge
k	Cercle
150 (ENTER)	Rayon = 150 unités
360 (ENTER)	Angle = 360 dégrés
c	Changement de couleur
b (ENTER)	bleu
g	Cercle plein
80 (ENTER)	Rayon = 80 unités
360 (ENTER)	Angle = 360 dégrés
s	Fin du programme

Le dessin ainsi réalisé n'est certainement pas une oeuvre d'art. Ce qui doit être ici représenté n'est d'ailleurs pas évident. C'est pourquoi nous laisserons le champ libre à vos interprétations...



7.3 Faire de la musique

"Enfin" diront les uns alors que les autres se demanderont où nous voulons en venir. Il s'agit en effet dans ce chapitre de musique programmée ou mieux de sons programmés.
La programmation de la musique est une application controversée parmi les utilisateurs d'ordinateurs. Jouer de la musique ne devrait-il pas plutôt être réservé aux hommes qui expriment ainsi leurs sentiments et leur vitalité? Nous renonçons volontairement à répondre ici à cette question. Rappelons simplement qu'entre autre de nombreux groupes pop font actuellement un grand usage des ordinateurs.

En tout cas, la programmation de la musique pourrait constituer un chapitre à part entière. Bien entendu il vous faut posséder déjà certaines connaissances en musique si vous voulez programmer de la musique; sinon, vous ne produirez que des sons épars ou même de simples bruits.

En tant que possesseur d'un CPC vous pouvez fort bien dépasser le niveau des bruits car les caractéristiques sonores et musicales du CPC sont excellentes. Vous pouvez ainsi, avec un peu de travail, imiter différents instruments et sortir la mélodie programmée sur vos haut-parleurs HIFI.

Le maniement des différentes instructions musicales n'est cependant pas tout simple. C'est pourquoi le programme de ce chapitre est surtout un programme d'exercice qui doit vous permettre de vous familiariser avec les instructions musicales. Vous pouvez cependant également utiliser le programme pour produire une mélodie avec peu de travail (par rapport au travail nécessité par la réalisation d'un programme de musique). Si vous n'avez pas encore d'expérience dans le domaine de la programmation de la musique, vous devez limiter vos ambitions et vous contenter de sortir différents sons et bruits. Ce n'est qu'ensuite que vous pourrez programmer une mélodie.

Ajoutons encore en ce qui concerne le programme qu'il n'est fondé que sur les trois instructions musicales de base SOUND (pour produire une note), ENV (pour définir une courbe d'enveloppe de volume) et ENT (pour définir une courbe d'enveloppe de hauteur de note).

Si vous voulez utiliser les autres fonctions et instructions musicales, il vous faut soit compléter le programme que nous vous présentons soit développer vous-même un autre programme musical.

Le programme est par ailleurs largement auto-commenté. Il offre à l'utilisateur les fonctions suivantes:

Instructions musicales

- entrer (seulement les différents paramètres)

- charger

- sauvegarder

- modifier (avec la possibilité d'examiner et de supprimer les instructions existantes).

- faire jouer de la musique

Les entrées pour les cinq sections des instructions de courbe d'enveloppe doivent être séparées par des virgules comme par exemple:

Nombre de pas, grandeur du pas, durée de la pause

Section 1 5,5,2 (ENTER)

Précisons que vous pouvez encore simplifier ce programme. Vous pouvez par exemple, lorsque des sections de programme se répètent, utiliser la technique des sous-programmes et par ailleurs vous pouvez également remplacer les variables p1(i)... p5(i) etc. par des variables doublement indicées en suivant le modèle suivant p(j,i) avec j = 1 ... 5 par exemple. Nous y avons renoncé ici pour obtenir une plus grande clarté du listing.
Nous avons également renoncé à intercepter les valeurs d'entrée non autorisées. Vous trouverez dans votre manuel d'utilisation l'explication détaillée des entrées possibles.

Encore un conseil pour finir: lorsque vous jouerez votre musique pour la première fois, il vaudrait peut-être mieux baisser un peu le niveau du volume.

```
10 REM I18
20 CLS
30 PRINT TAB(7)"I18 - Programme de musique"
40 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
50 PRINT TAB(9) "Bernd Kowal, 1985"
60 GOSUB 3230
70 DIM ks(500),tp(500),d(500),l(500),lh(500)
80 DIM th(500),gp(500),n1(15),p1(15),q1(15),r1(15)
90 DIM p2(15),q2(15),r2(15),p3(15),q3(15),r3(15)
100 DIM p4(15),q4(15),r4(15),p5(15),q5(15),r5(15)
110 DIM n2(15),t1(15),v1(15),w1(15),t2(15),v2(15)
120 DIM w2(15),t3(15),v3(15),w3(15),t4(15)
130 DIM v4(15),w4(15),t5(15),v5(15),w5(15)
140 REM
150 REM Menu
160 REM
170 PRINT TAB(17)"Menu":PRINT:PRINT
180 PRINT TAB(33)"Entree":PRINT
190 PRINT"Instructions musicales :":PRINT
200 PRINT" entrer";TAB(36)"1":PRINT
210 PRINT" charger";TAB(36)"2":PRINT
220 PRINT" corriger";TAB(36)"3":PRINT
230 PRINT" jouer";TAB(36)"4":PRINT
240 PRINT" sauvegarder";TAB(36)"5":PRINT
250 PRINT"Fin du travail";TAB(36)"6":PRINT:PRINT:PRINT
260 INPUT"Votre choix ";a:CLS
270 IF a<1 OR a>6 THEN CLS:GOTO 170
280 ON a GOTO 300,990,1300,2610,2820,3100
290 REM
300 REM Entree des instructions
310 REM musicales
320 PRINT TAB(17)"Menu":PRINT:PRINT
330 PRINT TAB(33)"Entree":PRINT
340 PRINT"Entree de":PRINT
350 PRINT" instructions SOUND";TAB(36)"1":PRINT
360 PRINT" instructions ENV";TAB(36)"2":PRINT
370 PRINT" instructions ENT";TAB(36)"3":PRINT
380 PRINT"Fin de l'entree";TAB(36)"4":PRINT:PRINT
390 PRINT:INPUT"Votre choix ";a:CLS
400 ON a GOTO 420,570,770,970
410 IF a<1 OR a>4 THEN 320
420 so=so+1
430 PRINT"Si vous entrez un 0 pour 'etat canal',"
440 PRINT"l'entree est terminee."
450 PRINT:PRINT
460 PRINT"L'instruction SOUND";so;":":PRINT
470 INPUT"Etat canal                  ";ks(so)
480 IF ks(so)=0 THEN so=so-1:CLS:GOTO 320
```

```
490 INPUT"Periode de la note       ";tp(so)
500 INPUT"Duree                    ";d(so)
510 INPUT"Volume                   ";l(so)
520 INPUT"Courbe enveloppe volume  ";lh(so)
530 INPUT"Courbe enveloppe note    ";th(so)
540 INPUT"Periode bruit            ";gp(so)
550 IF ver=1 THEN 1720
560 CLS:GOTO 420
570 ev=ev+1
580 PRINT"Si vous entrez un 0 comme 'numero de"
590 PRINT"courbe d'enveloppe', l'entree est"
600 PRINT"terminee.":PRINT:PRINT
610 PRINT"L'instruction ENV";ev;":":PRINT
620 INPUT"Numero de courbe d'enveloppe ";n1(ev):PRINT
630 IF n1(ev)=0 THEN ev=ev-1:CLS:GOTO 320
640 PRINT"Entrez (en separant par des virgules),"
650 PRINT"pour les 5 sections, le nombre de pas,"
660 PRINT"la grandeur du pas et la longueur de la"
665 PRINT"pause."
670 PRINT"       Nbre pas,grandeur pas,duree pause"
680 PRINT
690 PRINT"Section"
700 PRINT TAB(4) 1;"       ";:INPUT p1(ev),q1(ev),r1(ev)
710 PRINT TAB(4) 2;"       ";:INPUT p2(ev),q2(ev),r2(ev)
720 PRINT TAB(4) 3;"       ";:INPUT p3(ev),q3(ev),r3(ev)
730 PRINT TAB(4) 4;"       ";:INPUT p4(ev),q4(ev),r4(ev)
740 PRINT TAB(4) 5;"       ";:INPUT p5(ev),q5(ev),r5(ev)
750 IF ver=1 THEN 2140
760 CLS:GOTO 570
770 et=et+1
780 PRINT"Si vous entrez un 0 comme 'numero de"
790 PRINT"courbe d'enveloppe', l'entree est"
800 PRINT"terminee.":PRINT:PRINT
810 PRINT"L'instruction ENT";et;":":PRINT
820 INPUT"Numero de courbe d'enveloppe ";n2(et):PRINT
830 IF n2(et)=0 THEN et=et-1:CLS:GOTO 320
840 PRINT"Entrez (en separant par des virgules),"
850 PRINT"pour les 5 sections, le nombre de pas,"
860 PRINT"la grandeur du pas et la longueur de la"
865 PRINT"pause."
870 PRINT"       Nbre pas,grandeur pas,duree pause"
880 PRINT
890 PRINT"Section"
900 PRINT TAB(4) 1;"       ";:INPUT t1(et),v1(et),w1(et)
910 PRINT TAB(4) 2;"       ";:INPUT t2(et),v2(et),w2(et)
920 PRINT TAB(4) 3;"       ";:INPUT t3(et),v3(et),w3(et)
930 PRINT TAB(4) 4;"       ";:INPUT t4(et),v4(et),w4(et)
940 PRINT TAB(4) 5;"       ";:INPUT t5(et),v5(et),w5(et)
```

```
950 IF ver=1 THEN 2560
960 CLS:GOTO 770
970 CLS:GOTO 170
980 REM
990  REM Chargement des instructions
1000 REM musicales
1010 k1=so+1:k2=ev+1:k3=et+1
1020 OPENIN ""
1030 INPUT #9,k4,k5,k6
1040 k4=k4+so:k5=k5+ev:k6=k6+et
1050 FOR i=k1 TO k4
1060 INPUT #9,ks(i),tp(i),d(i),l(i),lh(i),th(i),gp(i)
1070 so=so+1
1080 NEXT i
1090 FOR i=k2 TO k5
1100 INPUT #9,n1(i)
1110 INPUT #9,p1(i),q1(i),r1(i)
1120 INPUT #9,p2(i),q2(i),r2(i)
1130 INPUT #9,p3(i),q3(i),r3(i)
1140 INPUT #9,p4(i),q4(i),r4(i)
1150 INPUT #9,p5(i),q5(i),r5(i)
1160 ev=ev+1
1170 NEXT i
1180 FOR i=k3 TO k6
1190 INPUT #9,n2(i)
1200 INPUT #9,t1(i),v1(i),w1(i)
1210 INPUT #9,t2(i),v2(i),w2(i)
1220 INPUT #9,t3(i),v3(i),w3(i)
1230 INPUT #9,t4(i),v4(i),w4(i)
1240 INPUT #9,t5(i),v5(i),w5(i)
1250 et=et+1
1260 NEXT i
1270 CLOSEIN
1280 CLS:GOTO 170
1290 REM
1300 REM Correction des instructions
1310 REM musicales
1320 PRINT TAB(17)"Menu":PRINT:PRINT
1330 PRINT TAB(33)"Entree":PRINT
1340 PRINT"Corrections de :":PRINT
1350 PRINT" instructions SOUND";TAB(36)"1":PRINT
1360 PRINT" instructions ENV";TAB(36)"2":PRINT
1370 PRINT" instructions ENT";TAB(36)"3":PRINT
1380 PRINT"Fin des corrections";TAB(36)"4":PRINT:PRINT
1390 PRINT:PRINT:PRINT
1400 PRINT:INPUT"Votre choix ";a:CLS
1410 ON a GOTO 1430,1750,2170,2590
1420 IF a<1 OR a>4 THEN 1320
```

```
1430 FOR i=1 TO so
1440 PRINT"L'instruction SOUND";i;"est :":PRINT
1450 INPUT"Etat canal                    ";ks(i)
1460 INPUT"Periode de la note            ";tp(i)
1470 INPUT"Duree                         ";d(i)
1480 INPUT"Volume                        ";l(i)
1490 INPUT"Courbe enveloppe volume ";lh(i)
1500 INPUT"Courbe enveloppe note   ";th(i)
1510 INPUT"Periode bruit                 ";gp(i)
1520 PRINT:PRINT:PRINT"Cette instruction doit-elle etre"
1530 PRINT:PRINT TAB(33)"Entree":PRINT
1540 PRINT"corrigee";TAB(36)"1"
1550 PRINT"ou"
1560 PRINT"supprimee";TAB(36)"2"
1570 PRINT"ou bien"
1580 PRINT"rester inchangee ?";TAB(36)"3":PRINT:PRINT
1590 INPUT"Votre choix ";a
1600 IF a<1 OR a>3 THEN 1520
1610 ON a GOTO 1690,1620,1730
1620 PRINT:PRINT:PRINT"Maintenant je supprime !"
1630 FOR j=i TO so-1
1640 ks(j)=ks(j+1):tp(j)=tp(j+1):d(j)=d(j+1)
1650 l(j)=l(j+1):lh(j)=lh(j+1)
1660 th(j)=th(j+1):gp(j)=gp(j+1)
1670 NEXT j
1680 so=so-1:CLS:GOTO 1320
1690 CLS:PRINT"Entrez donc maintenant les"
1700 PRINT"nouvelles valeurs :":PRINT
1710 ver=1:h=so:so=i:GOTO 470
1720 ver=0:so=h
1730 CLS:NEXT i
1740 CLS:GOTO 1320
1750 FOR i=1 TO ev
1760 PRINT"L'instruction ENV";i;"est :":PRINT
1770 PRINT"Numero de la courbe d'enveloppe :";n1(i):PRIN
T
1780 PRINT"       Nbre pas,grandeur pas,duree pause"
1790 PRINT
1800 PRINT"Section"
1810 PRINT TAB(4)1;TAB(11)p1(i);TAB(21)q1(i);
1820 PRINT TAB(31)r1(i)
1830 PRINT TAB(4)2;TAB(11)p2(i);TAB(21)q2(i);
1840 PRINT TAB(31)r2(i)
1850 PRINT TAB(4) 3;TAB(11) p3(i);TAB(21)q3(i);
1860 PRINT TAB(31)r3(i)
1870 PRINT TAB(4) 4;TAB(11) p4(i);TAB(21)q4(i);
1880 PRINT TAB(31)r4(i)
1890 PRINT TAB(4) 5;TAB(11) p5(i);TAB(21)q5(i);
```

```
1900 PRINT TAB(31)r5(i)
1910 PRINT:PRINT:PRINT"Cette instruction doit-elle etre"
1920 PRINT:PRINT TAB(33)"Entree":PRINT
1930 PRINT"corrigee";TAB(36)"1"
1940 PRINT"ou"
1950 PRINT"supprimee";TAB(36)"2"
1960 PRINT"ou bien"
1970 PRINT"rester inchangee ?";TAB(36)"3":PRINT
1980 INPUT"Votre choix ";a
1990 IF a<1 OR a>3 THEN 1910
2000 ON a GOTO 2110,2010,2150
2010 PRINT:PRINT:PRINT"Maintenant je supprime !"
2020 FOR j=i TO ev-1
2030 n1(j)=n1(j+1)
2040 p1(j)=p1(j+1):q1(j)=q1(j+1):r1(j)=r1(j+1)
2050 p2(j)=p2(j+1):q2(j)=q2(j+1):r2(j)=r2(j+1)
2060 p3(j)=p3(j+1):q3(j)=q3(j+1):r3(j)=r3(j+1)
2070 p4(j)=p4(j+1):q4(j)=q4(j+1):r4(j)=r4(j+1)
2080 p5(j)=p5(j+1):q5(j)=q5(j+1):r5(j)=r5(j+1)
2090 NEXT j
2100 ev=ev-1:CLS:GOTO 1320
2110 CLS:PRINT"Entrez donc maintenant les"
2120 PRINT"nouvelles valeurs :":PRINT
2130 ver=1:h=ev:ev=i:GOTO 620
2140 ver=0:ev=h
2150 CLS:NEXT i
2160 CLS:GOTO 1320
2170 FOR i=1 TO et
2180 PRINT"L'instruction ENT";i;"est :":PRINT
2190 PRINT"Numero de la courbe d'enveloppe :";n2(i):PRIN
T
2200 PRINT"        Nbre pas,grandeur pas,duree pause"
2210 PRINT
2220 PRINT"Section"
2230 PRINT TAB(4)1;TAB(11)t1(i);TAB(21)v1(i);
2240 PRINT TAB(31)w1(i)
2250 PRINT TAB(4)2;TAB(11)t2(i);TAB(21)v2(i);
2260 PRINT TAB(31)w2(i)
2270 PRINT TAB(4) 3;TAB(11) t3(i);TAB(21)v3(i);
2280 PRINT TAB(31)w3(i)
2290 PRINT TAB(4) 4;TAB(11) t4(i);TAB(21)v4(i);
2300 PRINT TAB(31)w4(i)
2310 PRINT TAB(4) 5;TAB(11) t5(i);TAB(21)v5(i);
2320 PRINT TAB(31)w5(i)
2330 PRINT:PRINT:PRINT"Cette instruction doit-elle etre"
2340 PRINT:PRINT TAB(33)"Entree":PRINT
2350 PRINT"corrigee";TAB(36)"1"
2360 PRINT"ou"
```

```
2370 PRINT"supprimee";TAB(36)"2"
2380 PRINT"ou bien"
2390 PRINT"rester inchangee ?";TAB(36)"3":PRINT
2400 INPUT"Votre choix ";a
2410 IF a<1 OR a>3 THEN 2330
2420 ON a GOTO 2530,2430,2570
2430 PRINT:PRINT:PRINT"Maintenant je supprime !"
2440 FOR j=i TO et-1
2450 n2(j)=n2(j+1)
2460 t1(j)=t1(j+1):v1(j)=v1(j+1):w1(j)=w1(j+1)
2470 t2(j)=t2(j+1):v2(j)=v2(j+1):w2(j)=w2(j+1)
2480 t3(j)=t3(j+1):v3(j)=v3(j+1):w3(j)=w3(j+1)
2490 t4(j)=t4(j+1):v4(j)=v4(j+1):w4(j)=w4(j+1)
2500 t5(j)=t5(j+1):v5(j)=v5(j+1):w5(j)=w5(j+1)
2510 NEXT j
2520 et=et-1:CLS:GOTO 1320
2530 CLS:PRINT"Entrez donc maintenant les"
2540 PRINT"nouvelles valeurs :":PRINT
2550 ver=1:h=et:et=i:GOTO 820
2560 ver=0:et=h
2570 CLS:NEXT i
2580 CLS:GOTO 1320
2590 CLS:GOTO 170
2600 REM
2610 REM Jouer un morceau de musique
2620 REM
2630 FOR i=1 TO ev
2640 p1=p1(i):q1=q1(i):r1=r1(i):p2=p2(i):q2=q2(i)
2650 r2=r2(i):p3=p3(i):q3=q3(i):r3=r3(i):p4=p4(i)
2660 q4=q4(i):r4=r4(i):p5=p5(i):q5=q5(i):r5=r5(i)
2670 n=n1(i)
2680 ENV n,p1,q1,r1,p2,q2,r2,p3,q3,r3,p4,q4,r4,p5,q5,r5
2690 NEXT i
2700 FOR i=1 TO et
2710 t1=t1(i):v1=v1(i):w1=w1(i):t2=t2(i):v2=v2(i)
2720 w2=w2(i):t3=t3(i):v3=v3(i):w3=w3(i):t4=t4(i)
2730 v4=v4(i):w4=w4(i):t5=t5(i):v5=v5(i):w5=w5(i)
2740 n=n2(i)
2750 ENT n,t1,v1,w1,t2,v2,w2,t3,v3,w3,t4,v4,w4,t5,v5,w5
2760 NEXT i
2770 FOR i=1 TO so
2780 SOUND ks(i),tp(i),d(i),l(i),lh(i),th(i),gp(i)
2790 NEXT i
2800 GOTO 170
2810 REM
2820 REM Sauvegarder des instructions
2830 REM musicales
2840 PRINT"Comment doit s'appeler le"
```

```
2850 INPUT"morceau de musique ";n$:PRINT
2860 OPENOUT n$
2870 PRINT #9,so,ev,et
2880 FOR i=1 TO so
2890 PRINT #9,ks(i),tp(i),d(i),l(i),lh(i),th(i),gp(i)
2900 NEXT i
2910 FOR i=1 TO ev
2920 PRINT #9,n1(i)
2930 PRINT #9,p1(i),q1(i),r1(i)
2940 PRINT #9,p2(i),q2(i),r2(i)
2950 PRINT #9,p3(i),q3(i),r3(i)
2960 PRINT #9,p4(i),q4(i),r4(i)
2970 PRINT #9,p5(i),q5(i),r5(i)
2980 NEXT i
2990 FOR i=1 TO et
3000 PRINT #9,n2(i)
3010 PRINT #9,t1(i),v1(i),w1(i)
3020 PRINT #9,t2(i),v2(i),w2(i)
3030 PRINT #9,t3(i),v3(i),w3(i)
3040 PRINT #9,t4(i),v4(i),w4(i)
3050 PRINT #9,t5(i),v5(i),w5(i)
3060 NEXT i
3070 CLOSEOUT
3080 CLS:GOTO 170
3090 REM
3100 REM Fin du Programme
3110 REM
3120 PRINT"Etes-vous sur de bien avoir sauvegarde"
3130 PRINT"les informations comme il convient ?"
3140 PRINT"Si ce n'est pas le cas, entrez"
3150 PRINT"simplement un 'oh' (sinon, la touche"
3160 PRINT"espace suffit)."
3170 INPUT a$
3180 IF a$="oh" THEN CLS:GOTO 2840
3190 PRINT:PRINT:PRINT"          Au revoir"
3200 PRINT:PRINT"Vous pouvez maintenant me debrancher"
3210 END
3220 REM
3230 REM SP Attendre
3240 REM
3250 LOCATE 5,25
3260 PRINT"Frappez une touche S.V.P"
3270 x$=INKEY$:IF x$="" THEN 3270
3280 CLS:RETURN
```

Liste de variables:

Les paramètres de l'instruction SOUND:

ks(i) = état canal

tp(i) = période de note

d(i) = durée

l(i) = volume

lh(i) = courbe d'enveloppe de volume

th(i) = courbe d'enveloppe de hauteur de note

gp(i) = période de bruit

Les paramètres de l'instruction ENV:

n1(i) = numéro de la courbe d'enveloppe

p1(i) ... p5(i) = nombre de pas (sections 1 ... 5)

q1(i) ... q5(i) = grandeur du pas (sections 1 ... 5)

r1(i) ... r5(i) = durée de la pause (sections 1 ... 5)

Les paramètres de l'instruction ENT:

n2(i) = numéro de la courbe d'enveloppe

t1(i) ... t5(i) = nombre de pas (sections 1 ... 5)

v1(i) ... v5(i) = grandeur du pas (sections 1 ... 5)

w1(i) ... w5(i) = durée de la pause (sections 1 ... 5)

Les autres variables:

a = réponse au menu

a$ = chaîne de caractères de réponse

et = nombre d'instructions ENT

ev = nombre d'instructions ENV

h = grandeur auxiliaire

i = index de comptage

j = index de comptage

k1
k2 } = valeurs initiales de la boucle
k3

k4
k5 } = valeurs finales de la boucle
k6

n$ = nom du morceau de musique

so = nombre d'instructions SOUND

ver = variable auxiliaire (égale 1 si des modifications sont apportées, sinon, égale à 0)

Description du programme:

Lignes 10-60 : Titre

Lignes 70-130 : Réservation de place mémoire pour 500 instructions SOUND et chaque fois 15 instructions ENV ou ENT

Lignes 140-280 : Menu

Lignes 290-970 : Entrée des instructions musicales
320-410 : Menu
420-560 : Paramètres de l'instruction SOUND
570-760 : Paramètres de l'instruction ENV
770-960 : Paramètres de l'instruction ENT
970 : Fin de l'entrée

Lignes 980-1280 : Chargement des instructions musicales
Les valeurs initiale et finale de boucle sont choisies de façon à ce qu'un chargement d'instructions supplémentaires soit également possible

Lignes 1290-2590 : Modification d'instructions musicales
1320-1420 : Menu
1430-1740 : Les instructions SOUND sont sorties sur l'écran. Elles peuvent être améliorées ou supprimées (menu lignes 1520-1600). La boucle de programme des lignes 1630-1670 permet la suppression d'une instruction SOUND. En cas de modification, la variable ver est fixée égale à 1 et on saute à la section de programme "entrer" (ligne 470, retour ligne 550).

Dans le tableau h est stockée la valeur actuelle de so car la variable so fonctionne comme index dans la section de programme "entrer".

1750-2580 : Le processus est identique pour les instructions ENT et ENV.

2590 : Fin de cette section de programme

Lignes 2600-2800 : Faire jouer les instructions musicales
Les instructions de courbe d'enveloppe sont exécutées en premier, suivies des instructions SOUND. Les paramètres indicés sont à cet effet chaque fois représentés sans index car sinon les variables de paramètres auraient un trop grand nombre de caractères ce qui entraînerait le message d'erreur "improper argument".

Lignes 2810-3080 : Sauvegarde des instructions musicales

Lignes 3090-3210 : Fin du programme
Il est cependant encore possible de sauvegarder les instructions musicales si on a oublié de le faire auparavant.

Lignes 3220-3290 : Sous-programme "attendre"

7.4 Le championnat de football

Le programme de ce chapitre gère les résultats du championnat de football et crée un tableau correspondant. Vous pouvez facilement adapter ce programme à d'autres sports. Il vous suffit pour cela de modifier les instructions DATA qui comportent la liste des clubs et éventuellement les calculs de classement pour la réalisation du tableau. Les instructions DATA doivent bien sûr également être modifiées à la fin de chaque saison pour tenir compte des clubs qui montent en première division ou qui descendent en seconde division.

Pour utiliser le programme, nous vous conseillons de vous conformer à la marche à suivre ci-dessous:

Etape 1: charger les résultats sauvegardés
Etape 2: entrer de nouveaux résultats
Etape 3: sauvegarder les anciens et les nouveaux résultats
Etape 4: réaliser le tableau

Vous pouvez en outre examiner à tout moment les résultats chargés ou entrés. Vous avez alors la possibilité de sortir sur l'écran tous les résultats d'une équipe unique.

Si l'entrée est effectuée avant le chargement des résultats, les résultats que vous venez d'entrer seront effacés. Cela peut cependant être évité, si vous le souhaitez, en modifiant ainsi la section de programme "charger les résultats":

```
95 DIM esp(18),eg(18),eu(18),ev(18),et1(18),et2(18)
1050 REM Charger les resultats
1060 REM
1070 OPENIN "Result."
1080 INPUT #9,eas
1090 FOR i=1 TO 20
1100 INPUT #9,esp(i),eg(i),eu(i),ev(i),et1(i),et2(i)
1105 sp(i)=sp(i)+esp(i):g(i)=g(i)+eg(i):u(i)=u(i)+eu(i)
1106 v(i)=v(i)+ev(i):t1(i)=t1(i)+et1(i):t2(i)=t2(i)+et2(
i)
1110 NEXT i
1120 FOR i=as+1 TO as+eas
1130 INPUT #9,s1$(i)
1140 INPUT #9,s2$(i)
1150 INPUT #9,st$(i)
1160 NEXT i:CLOSEIN
1165 as=as+eas
1170 CLS:GOTO 220
```

En lignes 1080 et 1100 sont créées de nouvelles variables (qui doivent évidemment être également dimensionnées) dont les valeurs seront ajoutées aux variables responsables du déroulement ultérieur du programme. En outre, les valeurs initiale et finale de la boucle en ligne 1120 sont modifiées pour empêcher l'effaçage de valeurs déjà existantes.

Cette version de programme présente cependant un inconvénient décisif: la fonction de programme "examiner" ne présente plus en effet les résultats dans l'ordre dans lequel ils ont été entrés. Si vous souhaitez notamment observer dans l'ordre chronologique les résultats d'une équipe bien précise, ou bien si vous voulez pouvoir dégager des tendances indiquant la force des différentes équipes au cours des différentes phases de la saison, il vaudra mieux que vous renonciez à cette version du programme.

Programme :

```
10 REM I19
20 CLS
30 PRINT "I19 - Programme de championnat de foot"
40 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
50 PRINT TAB(9) "Bernd Kowal, 1985"
60 GOSUB 1750
70 DIM bm$(20),k$(20),sp(20),g(20),u(20),v(20)
80 DIM t1(20),t2(20),p1(20),p2(20)
90 DIM s1$(380),s2$(380),st$(380)
100 DATA Paris SG,PSG,Toulon,To,Strasbourg,Str,Nantes,Na
110 DATA Rennes,Re,Bastia,Ba
120 DATA Monaco,Mo,Sochaux,So
130 DATA Metz,Me,Lens,Le
140 DATA Toulouse,Te,Nancy,Ny
150 DATA Lille,Li,Brest,Br
160 DATA Bordeaux,Bo,Nice,Ni
170 DATA Laval,La,Auxerre,Au
180 DATA Le Havre,Lh,Marseille,Ma
190 FOR i=1 TO 20
200 READ bm$(i),k$(i)
210 NEXT i
220 PRINT TAB(10)"Menu":PRINT:PRINT
230 PRINT TAB(33)"Entree":PRINT:PRINT
240 PRINT"Resultats :":PRINT
250 PRINT" entrer";TAB(36)"1":PRINT
260 PRINT" examiner";TAB(36)"2":PRINT
270 PRINT" charger";TAB(36)"3":PRINT
280 PRINT" sauvegarder (avant de"
290 PRINT" creer le tableau)";TAB(36)"4":PRINT
300 PRINT"Creer le tableau et"
310 PRINT"terminer le programme";TAB(36)"5":PRINT
320 PRINT:PRINT:INPUT"Votre choix ";a:CLS
330 IF a<1 OR a>5 THEN 220
340 ON a GOTO 360,700,1040,1180,1320
350 REM
360 REM Entree des resultats
370 REM
380 WINDOW #1,17,40,1,25
390 PRINT #1,"Pour l'entree, utilisez"
400 PRINT #1,"les abreviations"
410 PRINT #1,"suivantes :":PRINT #1
420 FOR i=1 TO 20
430 PRINT #1,bm$(i);"=";k$(i)
440 NEXT i
450 WINDOW #2,1,16,1,25
460 PRINT #2,"Combien de"
470 PRINT #2,"resultats"
```

```
480 PRINT #2,"faut-il"
490 INPUT #2,"entrer ";es:PRINT #2
500 FOR i=as+1 TO as+es
510 INPUT #2,"Locaux     ";ht$
520 INPUT #2,"Buts       ";th
530 INPUT #2,"Visiteurs  ";gt$
540 INPUT #2,"Buts       ";tg
550 FOR j=1 TO 20
560 IF ht$=k$(j) THEN j1=j
570 IF gt$=k$(j) THEN j2=j
580 NEXT j
590 sp(j1)=sp(j1)+1:sp(j2)=sp(j2)+1
600 IF th>tg THEN g(j1)=g(j1)+1:v(j2)=v(j2)+1
610 IF th=tg THEN u(j1)=u(j1)+1:u(j2)=u(j2)+1
620 IF th<tg THEN v(j1)=v(j1)+1:g(j2)=g(j2)+1
630 t1(j1)=t1(j1)+th:t2(j1)=t2(j1)+tg
640 t1(j2)=t1(j2)+tg:t2(j2)=t2(j2)+th
650 th$=STR$(th):tg$=STR$(tg)
660 s1$(i)=bm$(j1):s2$(i)=bm$(j2):st$(i)=th$+" :"+tg$
670 PRINT #2,:NEXT i:as=as+es
680 CLS:GOTO 220
690 REM
700 REM Examiner les resultats
710 REM
720 PRINT TAB(10)"Menu":PRINT:PRINT
730 PRINT TAB(33)"Entree":PRINT:PRINT
740 PRINT"Examiner tous les resultats";TAB(36)"1":PRINT
750 PRINT"Examiner les resultats"
760 PRINT"d'une equipe";TAB(36)"2":PRINT
770 PRINT"Fin de l'examen ";TAB(36)"3"
780 PRINT:PRINT:INPUT"Votre choix ";a:MODE 2
790 IF a<1 OR a>3 THEN MODE 1:GOTO 720
800 l=0:ON a GOTO 810,860,1020
810 FOR i=1 TO as
820 PRINT s1$(i);"-";s2$(i);TAB(45)st$(i)
830 l=l+1:IF l=20 THEN l=0:GOSUB 1750
840 NEXT i
850 GOSUB 1750:MODE 1:GOTO 720
860 PRINT"Pour l'entree, utilisez les"
870 PRINT"abreviations suivantes :":PRINT
880 FOR i=1 TO 20
890 PRINT bm$(i);"=";k$(i)
900 NEXT i:PRINT
910 INPUT"Quelle equipe ";m$:CLS
920 FOR i=1 TO 20
930 IF m$=k$(i) THEN m$=bm$(i)
940 NEXT i
950 FOR i=1 TO as
```

```
960 IF m$=s1$(i) OR m$=s2$(i) THEN 980
970 GOTO 1000
980 PRINT s1$(i);"-";s2$(i);TAB(45)st$(i)
990 l=l+1:IF l=20 THEN l=0:GOSUB 1750
1000 NEXT i
1010 GOSUB 1750:MODE 1:GOTO 720
1020 MODE 1:GOTO 220
1030 REM
1040 REM Charger resultats
1050 REM
1060 OPENIN "Result."
1070 INPUT #9,as
1080 FOR i=1 TO 20
1090 INPUT #9,sp(i),g(i),u(i),v(i),t1(i),t2(i)
1100 NEXT i
1110 FOR i=1 TO as
1120 INPUT #9,s1$(i)
1130 INPUT #9,s2$(i)
1140 INPUT #9,st$(i)
1150 NEXT i:CLOSEIN
1160 CLS:GOTO 220
1170 REM
1180 REM Sauvegarder resultats
1190 REM
1200 OPENOUT "Result."
1210 PRINT #9,as
1220 FOR i=1 TO 20
1230 PRINT #9,sp(i),g(i),u(i),v(i),t1(i),t2(i)
1240 NEXT i
1250 FOR i=1 TO as
1260 PRINT #9,s1$(i)
1270 PRINT #9,s2$(i)
1280 PRINT #9,st$(i)
1290 NEXT i:CLOSEOUT
1300 CLS:GOTO 220
1310 REM
1320 REM Creer le tableau
1330 REM
1340 FOR i=1 TO 20
1350 p1(i)=g(i)*2+u(i)
1360 p2(i)=v(i)*2+u(i)
1370 NEXT i
1380 FOR i=1 TO 19
1390 FOR j=i+1 TO 20
1400 IF p1(i)<p1(j) THEN GOSUB 1650
1410 IF p1(i)>p1(j) THEN 1470
1440 IF t1(i)-t2(i)<t1(j)-t2(j) THEN GOSUB 1650
1450 IF t1(i)-t2(i)>t1(j)-t2(j) THEN 1470
```

```
1460 IF t1(i)<t1(j) THEN GOSUB 1650
1470 NEXT j
1480 NEXT i
1490 MODE 2
1500 PRINT TAB(20)"Championnat de football de premiere d
ivision"
1510 PRINT:PRINT:PRINT TAB(30)"Mt.   g.   et   p.        but
s        Points"
1520 FOR i=1 TO 20
1530 PRINT USING "##";i;:PRINT". ";bm$(i);
1540 PRINT TAB(30) USING "##    ";sp(i);
1550 PRINT USING "##   ";g(i);u(i);v(i);
1560 PRINT USING "   ###   ";t1(i);
1570 PRINT":";:PRINT USING "### ";t2(i);
1580 PRINT USING "       ##   ";p1(i)
1590 REM
1600 NEXT i
1610 GOSUB 1750:MODE 1
1620 PRINT"Fin du programme":END
1630 REM
1640 REM Sous-programme d'echange
1650 h$=bm$(j):bm$(j)=bm$(i):bm$(i)=h$
1660 h=sp(j):sp(j)=sp(i):sp(i)=h
1670 h=g(j):g(j)=g(i):g(i)=h
1680 h=u(j):u(j)=u(i):u(i)=h
1690 h=v(j):v(j)=v(i):v(i)=h
1700 h=t1(j):t1(j)=t1(i):t1(i)=h
1710 h=t2(j):t2(j)=t2(i):t2(i)=h
1720 h=p1(j):p1(j)=p1(i):p1(i)=h
1730 h=p2(j):p2(j)=p2(i):p2(i)=h
1740 RETURN
1750 REM Sous-programme attendre
1760 LOCATE 5,25
1770 PRINT"Frappez une touche S.V.P. !"
1780 x$=INKEY$
1790 IF x$="" THEN 1780
1800 CLS:RETURN
```

Liste de variables:

a = réponse au menu

as = nombre de résultats de matchs

bm$(i) = équipes du championnat

es = nombre de résultats entrés

gt$ = équipe jouant à l'extérieur

g(i) = nombre de matchs gagnés par une équipe

h = variable auxiliaire du sous-programme "échanger"

h$ = variable auxiliaire du sous-programme "échanger"

ht$ = équipe locale

i = index de comptage

j = index de comptage

j1 = nombre index de l'équipe locale

j2 = nombre index de l'équipe invitée

k$(i) = abréviations des noms d'équipes du championnat

l = index de comptage

m$ = équipe dans la section de programme "examiner les résultats"

p1(i) = points positifs d'une équipe

p2(i) = points négatifs d'une équipe (inutilisé dans la version française du programme)

s1$(i)
= les équipes participant à un match de football
s2$(i)

sp(i) = nombre de matchs d'une équipe

st$(i) = résultat d'un match de football

t1(i) = buts marqués

t2(i) = buts encaissés

tg = buts marqués par l'équipe jouant à l'extérieur

tg$ = tg comme variable alphanumérique

th = buts de l'équipe recevant

th$ = th comme variable alphanumérique

u(i) = nombre de matchs nuls d'une équipe

v(i) = nombre de matchs perdus par une équipe

Description du programme:

Lignes 10-60 : Titre

Lignes 70-90 : Réservation de place mémoire pour 20 équipes et un total de 380 matchs

Lignes 100-180 : Les noms des équipes du championnat sont mis à la disposition du programme avec des abréviations correspondantes

Lignes 190-210 : Chargement des données

Lignes 220-340 : Menu et instructions de saut

Lignes 350-680 : Entrée des résultats de football

- 380-440 : Sortie d'une explication dans une fenêtre de texte définie
- 450-670 : Entrée des résultats dans le reste de l'écran. En lignes 550-580, les noms d'équipes entrés sont comparés aux données pour déterminer les index correspondant aux deux équipes participant au match. En fonction de ces index sont actualisées les variables responsables de la réalisation du tableau (lignes 590-640) et le résultat de football en tant que tel est placé dans les tableaux prévus à cet effet (lignes 650-660). A cet égard, la ligne 660 montre encore une fois une possibilité du traitement des chaînes de caractères.
- 680 : Retour au menu

Lignes 690-1020 : Examen des résultats

- 720-800 : Menu et branchements dans le programme
- 810-850 : Sortie de tous les résultats

860-1010 : Un nom d'équipe est réclamé (ligne 910), l'abréviation est identifiée (lignes 920-940) et sont sortis les résultats des matchs avec la participation de l'équipe indiquée (lignes 950-1000)

1020 : Retour au menu

Lignes 1030-1160 : Chargement des résultats

Lignes 1170-1300 : Sauvegarde des résultats

Lignes 1310-1610 : Réalisation du tableau

1340-1370 : Les points positifs et négatifs sont calculés pour chaque équipe

1380-1480 : Chaque équipe est comparée à toutes les autres et l'ordre est changé le cas échéant dans le sous-programme en lignes 1650-1740 (voyez également à ce sujet le chapitre 2.4 de cet ouvrage). Les valeurs utilisées pour la comparaison sont les points positifs (lignes 1400-1410), la différence de buts (lignes 1440-1450) et le nombre de buts marqués (ligne 1460). Ce n'est qu'en cas d'égalité pour un critère de comparaison qu'on passe à la comparaison avec le critère suivant.

1490-1610 : Sortie formatée d'un tableau

Ligne 1620 : Fin du programme

Lignes 1630-1740 : Sous-programme "échange"

Lignes 1750-1800 : Sous-programme "attendre"

7.5 Le puits à idées

- Recherche de rimes avec l'aide de l'ordinateur (les rimes correspondent en effet le plus souvent à une identité totale des dernières lettres de mots différents).

- Entretien de fleurs ou du jardin

- Fichier de recettes de cuisine

- Classer et gérer des collections de tous ordres

- Cours programmés (voile, morse, etc.)

- Evaluation des résultats de grandes manifestations sportives

- Plan d'entraînement avec l'ordinateur

- Durées d'exposition dans un laboratoire photo (votre CPC, avec son horloge en temps réel, est particulièrement approprié à une telle tâche).

- Graphisme en deux ou trois dimensions

- Programmation de patrons

- Plan de gestion des sections de lignes de votre chemin de fer (ou circuit) en modèle réduit.

- Commande informatisée de votre chemin de fer en modèle réduit (il suffit pour cela de disposer de quelques notions supplémentaires en électronique).

- Et bien d'autres choses encore.

Et voici comment pourrait se présenter un programme élémentaire d'entretien de vos fleurs.

```
10 REM Entretien des fleurs
20 k=4
30 REM k correspond au nombre de fleurs ou de plantes
40 DIM b$(k),w$(k),ls$(k)
50 DATA lis
51 DATA arroser peu
52 DATA garder a la lumiere et au chaud
60 DATA soleil
61 DATA arroser surtout par en bas
62 DATA garder a la lumiere et un peu au chaud
70 DATA dalhia
71 DATA arroser peu
72 DATA garder au frais et un peu a l'ombre
80 DATA begonia
81 DATA garder un peu au sec
82 DATA placer un peu a l'ombre
480 REM structure des instructions DATA
490 REM data fleur, eau, lumiere ou situation
500 FOR i=1 TO k
510 READ b$(i),w$(i),l$(i)
520 NEXT i
530 CLS
540 PRINT"Vous pouvez demander des informations"
550 PRINT"sur les fleurs ou plantes suivantes:"
560 FOR i=1 TO k
570 PRINT i;".  ";b$(i)
580 NEXT i
590 PRINT:PRINT:INPUT"Entrez un numero ";n
600 IF n>k OR n<1 THEN 590
610 CLS
620 PRINT b$(n):PRINT:PRINT:PRINT w$(n):PRINT:PRINT l$(n
)
630 PRINT:PRINT:PRINT:PRINT:PRINT "Pour d'autres informa
tions, veuillez"
640 PRINT "lancer le programme a nouveau"
```

Ce programme est construit comme le programme de calcul de calories. Il peut être aisément étendu à d'autres sortes de plantes. Une autre répartition des critères d'entretien est bien sûr également possible.

Nous vous proposerons encore un petit programme concernant le "tricot". Outre la possibilité de concevoir un pullover à l'écran et de faire imprimer des patrons de tricot au moyen d'une routine de hard copy, possibilité nécessitant techniquement beaucoup de travail de programmation, il est également possible, et c'est beaucoup plus simple à programmer, de réaliser un "calculateur de mailles". Il s'agit en effet simplement d'un problème de règle de trois:

```
10 REM Calculateur de mailles
20 CLS
30 INPUT "Nombre de mailles sur une largeur de 10 cm ";m
40 PRINT:PRINT:INPUT "Largeur du pullover en cm ";l
50 PRINT:PRINT:PRINT:PRINT
60 PRINT "Il vous faut donc ";m*l/10;" mailles."
```

8. Ces chers petits

8.1 Remarque préalable

Ce chapitre a pour but de vous montrer comment un écolier ou ùn étudiant peuvent employer l'AMSTRAD CPC. Vous avez peut-être des enfants scolarisés auxquels vous pourrez présenter les programmes de ce chapitre, un peu comme un contre-poids aux programmes de jeu largement répandus parmi les écoliers.

Il est évidemment possible de développer des programmes très utiles pour les problèmes les plus différents qui meublent le quotidien de la vie scolaire. Les problèmes mathématiques occupent cependant la première place, les mathématiques étant certainement la matière la plus "proche" des ordinateurs. Vous trouverez cependant également un programme de vocabulaire un peu plus ambitieux dont l'intérêt est notamment de pouvoir être aisément modifié pour être adapté à d'autres questions.

8.2 Table de multiplication

Le programme de ce chapitre conviendra particulièrement à l'écolier en primaire. Il peut être modifié sans problème pour fonctionner avec une autre opération arithmétique de base. Une partie considérable de l'arithmétique élémentaire peut donc être ainsi couverte par ce programme.

A partir d'une valeur maximale à entrer, des nombres aléatoires sont produits dans le déroulement du programme qui serviront à poser un problème de multiplication. Cet exercice doit alors être résolu aussi correctement que possible. Une fois les exercices terminés, le taux de réussite est indiqué. Il s'agit donc, comme pour l'entraînement à l'apprentissage du vocabulaire, d'un programme d'interrogation.

Programme :

```
10 REM I20
20 CLS
30 PRINT "I20 - Programme d'exercice"
40 PRINT TAB(7)"de table de multiplication"
50 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
60 PRINT TAB(9) "Bernd Kowal, 1985"
70 GOSUB 350
80 PRINT"En entrant la lettre 'z', vous pouvez"
90 PRINT"interrompre l'exercice suivant"
100 PRINT:PRINT"Mais veuillez entrer d'abord le"
110 PRINT"nombre maximum que peut atteindre"
120 INPUT"l'exercice ";qz:CLS
130 i=i+1
140 PRINT"Exercice";i;":":PRINT:PRINT:PRINT
150 f1=ROUND(RND*qz):f2=ROUND(RND*qz)
160 PRINT TAB(10) f1;"fois";f2;:INPUT" =  ",ep$
170 IF ep$="z" THEN 250
180 ep=VAL(ep$)
190 p=f1*f2:PRINT:PRINT
200 IF ep<>p THEN 220
210 PRINT TAB(11)"juste":ra=ra+1:GOTO 240
220 PRINT TAB(11)"faux":fa=fa+1:PRINT:PRINT
230 PRINT TAB(10) f1;"fois";f2;" = ";p
240 GOSUB 350:GOTO 130
250 CLS:i=i-1
260 PRINT" Sur";i;"exercices, ":PRINT
270 PRINT fa;"reponses etaient fausses et":PRINT
280 PRINT ra;"reponses etaient justes":PRINT
290 PRINT:PRINT:PRINT:PRINT
300 PRINT ROUND(fa/i*100,2);
310 PRINT"% des reponses etaient donc fausses"
320 PRINT:PRINT" et";ROUND(ra/i*100,2);
330 PRINT" % etaient justes."
340 PRINT:PRINT:PRINT"Fin des calculs":END
350 REM Sous-programme attendre
360 LOCATE 5,25
370 PRINT"Frappez une touche S.V.P. !"
380 x$=INKEY$
390 IF x$="" THEN 380
400 CLS:RETURN
```

Liste de variables:

ep = produit entré

ep$ = tableau d'entrée pour ep

f1 = premier facteur

f2 = second facteur

fa = nombre de réponses fausses

gz = nombre maximum de l'exercice

i = nombre de problèmes posés

p = produit calculé

ra = nombre de réponses correctes

Description du programme:

Lignes 10-70 :	Titre
Lignes 80-120 :	Sortie d'une explication de l'utilisation du programme et demande d'une valeur pour le nombre maximum de l'exercice
Lignes 130-160 :	Un problème est posé et un résultat est demandé
Lignes 170-180 :	Si l'exercice doit être interrompu, on saute à la sortie du taux de réussite. Sinon l'entrée effectuée est convertie en une valeur numérique
Lignes 190-240 :	On teste la justesse de l'entrée, un commentaire en conséquence est sorti et le résultat de la vérification est enregistré. On saute ensuite à la ligne 130 pour poser un autre problème.
Lignes 250-330 :	Sortie du taux de réussite absolu et relatif
Lignes 340 :	Fin du programme
Lignes 350-400 :	Sous-programme "attendre"

8.3 Test de nombres premiers

Un nombre entier x est dit nombre premier s'il ne peut être divisé sans reste que par lui-même et par 1. Le nombre 1 constitue une exception puisqu'il n'est pas considéré comme premier bien que ne pouvant être divisé que par 1.

Si l'on excepte le nombre 2, qui n'est pas habituellement considéré comme un vrai nombre premier, on peut constater par ailleurs qu'un nombre pair ne peut être un nombre premier puisqu'il est par définition divisible par 2.

Le test des nombres pairs peut cependant prendre un certain temps. Ils doivent en effet être divisés par tous les diviseurs supérieurs à 1 mais inférieurs à eux-mêmes. Si une de ces divisions produit un résultat entier (division sans reste), le nombre testé n'est pas un nombre premier. Ce n'est que s'il y a un reste pour toutes les divisions effectuées qu'on est en présence d'un nombre premier.

Il est évident que le test de grands nombres prend énormément de temps si l'on ne dispose pas d'un ordinateur. Par ailleurs le programme nécessaire pour ce travail n'est pas très difficile. Le déroulement du programme découle directement des explications précédentes.

Programme:

```
10 REM I21
20 CLS
30 PRINT "I21 - Programme de test des"
40 PRINT TAB(7)"nombres premiers"
50 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
60 PRINT TAB(9) "Bernd Kowal, 1985"
70 GOSUB 210
80 PRINT"Entrez un nombre entier"
90 INPUT"positif ";z
100 PRINT:PRINT:PRINT
110 IF z<>1 THEN 140
120 PRINT"Le nombre 1 ne fait pas partie"
130 PRINT"des nombres premiers.":GOTO 200
140 FOR i=2 TO z/2
150 IF z/i=INT(z/i) THEN 180
160 NEXT i
170 PRINT z;"est un nombre premier.":GOTO 200
180 PRINT z;"n'est pas un nombre premier car"
190 PRINT z;"est divisible par";i
200 PRINT:PRINT:PRINT"Fin des calculs":END
210 REM Sous-programme attendre
220 LOCATE 5,25
230 PRINT"Frappez une touche S.V.P. !"
240 x$=INKEY$
250 IF x$="" THEN 240
260 CLS:RETURN
```

Liste de variables:

i = diviseur

z = nombre à tester

Description du programme:

Lignes 10-70 : Titre

Lignes 80-90 : Entrée d'un nombre

Ligne 100 : Impression de trois lignes vides

Lignes 110-130 : Si le nombre 1 a été entré, un commentaire est sorti et on saute à la fin du programme

Lignes 140-160 : Dans cette boucle de programme, le diviseur i est chaque fois augmenté de 1 jusqu'à ce qu'ait été atteinte la valeur z/2. Si une division sans reste s'est produite, au saute à la sortie du résultat correspondant. La boucle ne tourne que jusqu'à z/2 car avec des nombres supérieurs il ne serait plus possible de toute façon d'arriver à un résultat entier.
Pour z=2 et z=3, la boucle n'est pas parcourue puisque z/2<2. On atteint toutefois la partie qui convient de la sortie du résultat en ligne 170.

Lignes 170-190 : Sortie du résultat

Ligne 200 : Fin du programme

Lignes 210-260 : Sous-programme "attendre"

8.4 Détermination du point zéro

Le calcul du point zéro pour des fonctions quelconques constitue une partie importante des mathématiques. De nombreux problèmes mathématiques lui sont liés.

Le procédé qui est à la base du programme que nous vous proposons est celui de l'imbrication d'intervalles. A l'intérieur d'un intervalle de limites a et b, figure obligatoirement un point zéro si les signes des valeurs de fonction f(a) et f(b) sont différents. Si le signe de f($\frac{a+b}{2}$) est égal au signe de f(a), alors le point zéro se trouve dans la partie droite de l'intervalle. Dans ce cas on affecte la valeur $\frac{a+b}{2}$ à a et on recommence l'opération. De même, on affectera à b la valeur $\frac{a+b}{2}$ si le point zéro se trouve dans la partie gauche de l'intervalle, c'est-à-dire si le signe de f($\frac{a+b}{2}$) est égal au signe de f(b).

Cette procédure pourrait être répétée pratiquement à l'infini. C'est pourquoi il faut fournir une valeur, comme critère d'interruption, qui limitera le nombre d'imbrications d'intervalles. Dans notre programme, cette valeur est de 500 (ligne 270). Bien sûr pour de petits intervalles, un nombre inférieur d'imbrications suffirait pour obtenir un résultat précis mais cette valeur peut par contre se révéler trop petite pour les très grands intervalles.

Le programme que nous vous présentons suppose que les valeurs de fonction des limites de l'intervalle soient de signe différent bien qu'il puisse également y avoir des points zéro dans un intervalle si cette condition n'est pas remplie. Cela signifie simplement dans ce cas qu'un trop grand intervalle a été choisi. Par ailleurs, nous avons négligé le cas particulier de l'existence de plusieurs points zéro. C'est pourquoi il est conseillé de travailler avec des intervalles relativement petits.

Vous pouvez appliquer ce programme à n'importe quelles fonctions. Il suffit de modifier chaque fois en conséquence la ligne de programme 340. Faites cependant attention à la régularité de la fonction entrée, sous peine de provoquer des messages d'erreur. Dans notre programme, la fonction prescrite est la fonction simple:

y = x^3 - 5

Cette fonction a un point zéro pour

x = 1.7099 ...

Programme:

```
10 REM I22
20 CLS
30 PRINT "I22 - Programme pour determiner"
40 PRINT TAB(7)"les points zero"
50 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
60 PRINT TAB(9) "Bernd Kowal, 1985"
70 GOSUB 360
80 PRINT"La ligne 340 definit une fonction fixe."
90 PRINT"En modifiant cette ligne du programme,"
100 PRINT"vous pouvez utiliser le programme"
110 PRINT"egalement pour une autre fonction"
120 GOSUB 360
130 PRINT"Quel intervalle faut-il"
140 PRINT"examiner ?":PRINT
150 INPUT"Debut de l'intervalle a ";a
160 INPUT"Fin de l'intervalle   b ";b
170 x=a:GOSUB 340:fa=y
180 x=b:GOSUB 340:fb=y
190 PRINT:PRINT:PRINT"f(a) =";fa
200 PRINT"f(b) =";fb:PRINT:PRINT
210 IF fa=0 THEN PRINT"Point zero pour x =";a:GOTO 330
220 IF fb=0 THEN PRINT"Point zero pour x =;b:GOTO 330
230 IF SGN(fa)<>SGN(fb) THEN 270
240 PRINT"La condition, signe de f(a)<>"
250 PRINT"signe f(b) n'est pas remplie."
260 GOTO 330
270 FOR i=1 TO 500
280 x=(a+b)/2:GOSUB 340
290 IF SGN(y)=SGN(fa) THEN a=x:fa=y ELSE b=x:fb=y
300 NEXT i
310 PRINT:PRINT:PRINT"Point zero pour x =";x
320 PRINT"f(x) =";y
330 PRINT:PRINT:PRINT"Fin des calculs":END
340 y=x^3-5
350 RETURN
360 REM Sous-programme attendre
370 LOCATE 5,25
380 PRINT"Frappez une touche S.V.P. !"
390 x$=INKEY$
400 IF x$="" THEN 390
410 CLS:RETURN
```

Liste de variables:

a = début de l'intervalle

b = fin de l'intervalle

fa = valeur de fonction du début de l'intervalle

fb = valeur de fonction de la fin de l'intervalle

i = index de comptage

x = valeur x de la fonction

y = valeur y de la fonction

Description du programme:

Lignes 10-70 :	Titre
Lignes 80-120 :	Sortie d'une explication et "attendre"
Lignes 130-160 :	Entrée des limites de l'intervalle
Lignes 170-200 :	Les valeurs de fonction des limites de l'intervalle sont calculées et sorties par des sauts au sous-programme (ligne 340)
Lignes 210-220 :	Sortie d'un commentaire et saut à la fin du programme si la valeur de fonction d'une limite d'intervalle est égale à 0

Lignes 230-260 : Si la condition de signes différents n'est pas remplie un commentaire est sorti et on saute à la fin du programme

Ligne 270 : Ouverture d'une boucle de programme pour 500 imbrications d'intervalles

Ligne 280 : L'intervalle est divisé par deux et la valeur de fonction correspondante est calculée

Ligne 290 : Si le signe de la valeur de fonction du milieu de l'intervalle SGN(y) est égal au signe de la valeur de fonction de la limite gauche de l'intervalle SGN(fa), alors le milieu de l'intervalle devient la limite gauche de l'intervalle pour le suivant intervalle divisé par deux et la valeur de fonction du milieu de l'intervalle devient la valeur de fonction de la limite gauche de l'intervalle suivant. Si le signe de la valeur de fonction du milieu de l'intervalle SGN(y) est différent du signe de la valeur de fonction de la limite gauche de l'intervalle SGN(fa), alors le milieu de l'intervalle devient la limite droite de l'intervalle pour le suivant intervalle divisé par deux et la valeur de fonction du milieu de l'intervalle devient la valeur de fonction de la limite droite de l'intervalle suivant.

Ligne 300 : Fin de la boucle de programme

Lignes 310-320 : Sortie du résultat

Ligne 330 : Fin du programme

Lignes 340-350 : Sous-programme de définition de la fonction

Lignes 360-410 : Sous-programme "attendre"

8.5 Entraînement au vocabulaire

Dans l'apprentissage de langues étrangères, votre CPC peut également vous fournir une aide efficace. Il serait par exemple possible de développer un programme entier d'enseignement d'une langue étrangère. Cela dépasserait cependant le cadre de cet ouvrage. Avec de bonnes ou très bonnes connaissances en BASIC, vous pourriez vous y essayer vous-même. Il vous faudrait bien sûr également de solides connaissances linguistiques et surtout une profonde motivation pour ce type de travail.

Dans ce chapitre, il s'agit d'une interrogation au tac-au-tac sur le vocabulaire. Vous avez certainement l'habitude d'apprendre du vocabulaire en cachant une colonne de votre lexique. Cette méthode présente l'inconvénient que l'ordre des mots est toujours le même. Si lors de l'interrogation sur le vocabulaire un ordre différent est choisi -et c'est certainement le cas-, vous aurez du mal à avoir autant de réussite "qu'à la maison". Il vous faut donc quelqu'un qui ait du temps à vous consacrer et qui puisse travailler avec vous. Cela pourrait être désormais votre CPC. Il sera toujours disponible pour vous assister. Vous trouverez dans les pages suivantes le programme nécessaire. Il est conçu d'une façon tellement pratique qu'il peut prendre en charge toute l'organisation de l'apprentissage du vocabulaire.

Le programme dispose des fonctions suivantes:

- entrée du vocabulaire
- chargement du vocabulaire
- correction du vocabulaire (avec la possibilité d'examiner tous les mots figurant dans la mémoire de travail ainsi que celle de supprimer des mots).

- compléter le vocabulaire (au clavier ou par un chargement supplémentaire).

- apprentissage du vocabulaire (mot français - mot étranger ou mot étranger - mot français).

- sauvegarde du vocabulaire

Si 50 mots se trouvent par exemple dans la mémoire de travail de l'ordinateur, 50 mots seront d'abord choisis au hasard. Les mots pour lesquels une réponse positive a été donnée ne sont pas proposés une seconde fois mais les mots pour lesquels la réponse a été incorrecte peuvent être présentés plusieurs fois. Ensuite vient le tour de tous les mots qui n'ont pas encore été choisis par l'ordinateur. On répète enfin l'interrogation des mots pour lesquels une réponse incorrecte a été entrée jusqu'à ce qu'une réponse correcte ait été fournie au moins une fois pour chaque mot.

Il est bien sûr également possible de mettre fin prématurément à l'exercice. En tout cas un taux de réussite est indiqué à la fin de l'exercice.

Pour ce travail d'apprentissage, il est conseillé de sauvegarder séparément les différentes leçons. Vous pouvez ensuite les réunir avec la fonction de complément. Le programme réserve de la place en mémoire pour un nombre global de 1000 mots.

Pour le travail sur disquette, une légère modification du programme s'impose:

```
565 INPUT "Comment s'appelle la leçon ";n$
570 OPENIN n$
```

L'entrée du vocabulaire s'effectue en entrant d'abord le mot français, suivi d'une virgule, puis le mot étranger. Exemple:

? il,he (ENTER)

Il faut noter que les compléments ne peuvent être effectués qu'avec la fonction "compléter". Le fait d'entrer ou de charger deux fois du vocabulaire aurait sinon pour effet d'effacer les mots figurant dans la mémoire du programme.
Le programme est par ailleurs très largement auto-explicatif.

Le mieux est donc que vous l'essayiez tout de suite!

```
10 REM I23
20 CLS
30 PRINT TAB(7)"I23 - Programme de vocabulaire"
40 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
50 PRINT TAB(9) "Bernd Kowal, 1985"
60 GOSUB 2280
70 DIM d$(1000),f$(1000),w(1000),g(1000)
80 k1=1:h=15
90 REM
100 REM Menu
110 REM
120 PRINT TAB(17)"Menu":PRINT:PRINT
130 PRINT TAB(33)"Entree":PRINT
140 PRINT"Vocabulaire  :":PRINT
150 PRINT" entree";TAB(36)"1":PRINT
160 PRINT" chargement";TAB(36)"2":PRINT
170 PRINT" correction";TAB(36)"3":PRINT
180 PRINT" completer";TAB(36)"4":PRINT
190 PRINT" apprendre";TAB(36)"5":PRINT
200 PRINT" sauvegarder";TAB(36)"6":PRINT
210 PRINT"Fin du travail";TAB(36)"7":PRINT:PRINT:PRINT
220 INPUT"Votre choix ";a:CLS
230 IF a<1 OR a>7 THEN CLS:GOTO 120
240 ON a GOTO 370,540,670,260,1390,1880,1980
250 REM
260 REM Completer le vocabulaire
270 REM
280 PRINT"Faut-il charger du vocabulaire"
290 INPUT"supplementaire (o/n) ";a$
300 IF a$="o" THEN e=k1+1:GOTO 570
310 IF a$<>"n" THEN 280
320 PRINT:PRINT
330 PRINT"Vous pouvez donc entrer maintenant du"
340 PRINT"vocabulaire supplementaire.":GOSUB 2280
350 k1=k1+1:h=k1+15
360 REM
370 REM Entrer du vocabulaire
380 REM
390 PRINT"Les mots francais et etranger doivent"
400 PRINT"etre separes entre eux par une virgule."
410 PRINT"Entrez *,* lorsque tout le vocabulaire"
420 PRINT"aura ete tape.":PRINT
430 PRINT"Mot francais, mot etranger":PRINT
440 FOR i=k1 TO 1000
450 INPUT d$(i),f$(i)
460 IF d$(i)<>"*" THEN 500
470 k1=i-1:PRINT:PRINT"Il y a maintenant";k1
480 PRINT"mots dans la memoire de travail."
```

```
490 GOSUB 2280:GOTO 120
500 IF i=h THEN k1=h+1:h=h+15:GOSUB 2280:GOTO 390
510 NEXT i
520 GOSUB 2280
530 REM
540 REM Charger du vocabulaire
550 REM
560 e=1
570 OPENIN ""
580 FOR i=e TO 1000
590 INPUT #9,d$(i):INPUT #9,f$(i)
600 IF EOF=-1 THEN k1=i:GOTO 620
610 NEXT i
620 CLOSEIN
630 PRINT:PRINT"Il y a maintenant";k1;
640 PRINT"mots dans la memoire de travail."
650 GOSUB 2280:GOTO 120
660 REM
670 REM Correction du vocabulaire
680 REM
690 PRINT"Pouvez-vous indiquer le mot incorrect"
700 PRINT"ou preferez-vous examiner"
710 PRINT"tout le vocabulaire ?":PRINT
720 PRINT TAB(20)"Entree":PRINT
730 PRINT"indiquer";TAB(23)"1":PRINT
740 PRINT"examiner";TAB(23)"2"
750 PRINT:INPUT"Votre choix ";a:CLS
760 IF a=2 THEN 920
770 IF a<>1 THEN 690
780 PRINT"Veuillez entrer le mot"
790 INPUT"incorrect ";fw$
800 INPUT"Quelle est l'expression juste ";rw$
810 FOR i=1 TO k1
820 IF d$(i)=fw$ THEN d$(i)=rw$:GOTO 870
830 IF f$(i)=fw$ THEN f$(i)=rw$:GOTO 870
840 NEXT i
850 PRINT:PRINT"Le mot indique est introuvable"
860 GOTO 880
870 PRINT:PRINT"La correction a ete effectuee."
880 PRINT:INPUT"Encore une correction (o/n) ";a$
890 IF a$="n" THEN CLS:GOTO 120
900 IF a$<>"o" THEN 880
910 CLS:GOTO 690
920 WINDOW #1,20,40,1,25
930 k2=1:k3=10
940 IF k2<=k1 THEN 1040
950 FOR i=1 TO k1
960 IF d$(i)<>"" THEN 1020
```

```
970 k1=k1-1
980 FOR j=i TO k1
990 d$(j)=d$(j+1)
1000 f$(j)=f$(j+1)
1010 NEXT j
1020 NEXT i
1030 CLS:GOTO 120
1040 FOR i=k2 TO k3
1050 IF i>k1 THEN 1090
1060 PRINT #1,i;". ";d$(i);"-"
1070 PRINT #1,TAB(6)f$(i)
1080 NEXT i
1090 PRINT"Ou est"
1100 PRINT"l'erreur ?"
1110 PRINT"Entrez un 0"
1120 PRINT"si tout est"
1130 PRINT"correct."
1140 PRINT"Le fait d'entrer"
1150 PRINT"un espace pour le"
1160 PRINT"mot francais"
1170 PRINT"entraine la"
1180 PRINT"suppression du"
1190 PRINT"numero"
1200 PRINT"correspondant.":PRINT
1210 INPUT"Numero ";b
1220 IF b=0 THEN 1370
1230 PRINT:PRINT"Le mot francais"
1240 PRINT"est-il incorrect"
1250 INPUT"(o/n) ";a$
1260 IF a$="n" THEN 1300
1270 IF a$<>"o" THEN 1230
1280 PRINT"Quelle est l'ex-
1290 INPUT"pression correcte ";d$(b)
1300 PRINT:PRINT"Le mot etranger"
1310 INPUT"est-il faux (o/n) ";a$
1320 IF a$="n" THEN 1360
1330 IF a$<>"o" THEN 1300
1340 PRINT"Quelle est l'ex-
1350 INPUT"pression correcte ";f$(b)
1360 CLS:CLS #1:GOTO 1040
1370 CLS:CLS #1:k2=k3+1:k3=k3+10:GOTO 940
1380 REM
1390 REM Apprentissage du vocabulaire
1400 REM
1410 l=0:rr=0:ff=0
1420 FOR i=1 TO k1:q(i)=0:w(i)=0:NEXT i
1430 PRINT"Il y a";k1;"mots dans l'exercice."
1440 PRINT:PRINT"Entrez une * pour le mot demande"
```

```
1450 PRINT"lorsque vous voudrez interrompre"
1460 PRINT"l'exercice.":PRINT
1470 PRINT:PRINT"Comment les mots doivent-ils etre"
1480 PRINT"demandes ?"
1490 PRINT:PRINT TAB(33)"Entree":PRINT
1500 PRINT"Mot francais - mot etranger";TAB(36)"1"
1510 PRINT
1520 PRINT"Mot etranger - mot francais";TAB(36)"2"
1530 PRINT:INPUT"Votre choix ";c
1540 IF c<1 OR c>2 THEN 1470 ELSE CLS
1550 y=INT(RND*k1+1)
1560 IF w(y)=1 THEN 1550
1570 GOSUB 2110
1580 l=l+1:q(y)=1
1590 IF l<k1 THEN 1550
1600 FOR i=1 TO k1
1610 IF q(i)=1 THEN 1630
1620 y=i:GOSUB 2110:l=l+1
1630 NEXT i
1640 hi=0
1650 FOR i=1 TO k1
1660 IF w(i)=1 THEN 1690
1670 y=i:GOSUB 2110:l=l+1
1680 IF w(y)=0 THEN hi=1
1690 NEXT i
1700 IF hi=1 THEN 1640
1710 PRINT"Vous avez maintenant repondu juste"
1720 PRINT"au moins une fois pour tous les mots.":PRINT
1730 PRINT:PRINT"Sur un total de";l;"mots demandes"
1740 PRINT rr;"reponses etaient correctes"
1750 PRINT"et";ff;"fausses.":PRINT:PRINT
1760 PRINT"Soit en pourcentage :":PRINT
1770 PRINT; USING "###.##";rr*100/l;
1780 PRINT"% des reponses etaient correctes"
1790 PRINT"et"
1800 PRINT; USING "###.##";ff*100/l;
1810 PRINT"% des reponses etaient fausses"
1820 PRINT:PRINT: IF ff<=rr THEN 1840
1830 PRINT"Vous devez donc encore travailler":PRINT"un p
eu."
1840 IF ff*10>rr THEN 1860
1850 PRINT"C'est un bon resultat."
1860 GOSUB 2280:GOTO 120
1870 REM
1880 REM Sauvegarde du vocabulaire
1890 REM
1900 PRINT"Comment doit s'appeler la lecon"
1910 INPUT"a sauvegarder ";n$:PRINT
```

```
1920 OPENOUT n$
1930 FOR i=1 TO k1
1940 PRINT #9,d$(i):PRINT #9,f$(i)
1950 NEXT i:CLOSEOUT
1960 CLS:GOTO 120
1970 REM
1980 REM Fin du Programme
1990 REM
2000 PRINT"Etes-vous sur de bien avoir sauvegarde"
2010 PRINT"les informations comme il convient ?"
2020 PRINT"Si ce n'est pas le cas, entrez"
2030 PRINT"simplement un 'oh' (sinon, la touche"
2040 PRINT"espace suffit)."
2050 INPUT a$
2060 IF a$="oh" THEN CLS:GOTO 1900
2070 PRINT:PRINT:PRINT"          Au revoir"
2080 PRINT:PRINT"Vous pouvez maintenant me debrancher"
2090 END
2100 REM
2110 REM SP Interrogation
2120 REM
2130 LOCATE 8,4
2140 IF c=2 THEN 2170
2150 PRINT d$(y);"  ";:INPUT p$
2160 IF p$=f$(y) THEN 2190 ELSE 2210
2170 PRINT f$(y);"  ";:INPUT p$
2180 IF p$<>d$(y) THEN 2210
2190 PRINT:PRINT:PRINT TAB(9)"juste"
2200 rr=rr+1:w(y)=1:GOTO 2260
2210 PRINT:IF p$="*" THEN CLS:GOTO 1730
2220 PRINT:PRINT TAB(9)"faux":ff=ff+1:w(y)=0
2230 PRINT:PRINT:PRINT TAB(9)"la bonne reponse est :"
2240 PRINT:IF c=1 THEN PRINT TAB(9)f$(y)
2250 IF c=2 THEN PRINT TAB(9)d$(y)
2260 GOSUB 2280:RETURN
2270 REM
2280 REM SP Attendre
2290 REM
2300 LOCATE 7,25
2310 PRINT"Frappez une touche S.V.P"
2320 x$=INKEY$
2330 IF x$="" THEN 2320
2340 CLS:RETURN
```

Liste de variables:

a = réponse au menu

a$ = chaîne de caractères de réponse (o/n)

b = variable auxiliaire et index (section de programme "corrections")

c = réponse au menu dans la section de programme "apprendre"

d$(i) = mot français

e = valeur initiale de la boucle (supérieure à 1 si l'on complète le vocabulaire)

f$(i) = mot étranger

ff = nombre de réponses fausses

fw$ = mot incorrect (section de programme "corrections")

h = variable auxiliaire qui indique si l'écran est plein

hi = variable auxiliaire (égale 1 s'il manque encore des réponses justes)

i = index de comptage

j = index de comptage

k1 = nombre de mots dans la mémoire de travail

k2 = début de la boucle

k3 = fin de la boucle

l = nombre total de réponses

$ = nom de la leçon à sauvegarder

p$ = entrée dans le sous-programme "interrogation"

q(i) = variable auxiliaire (égale 0 si un mot n'a pas encore été demandé, sinon égale 1)

r = nombre de réponses justes

rw$ = mot correct (section de programme "corrections")

w(i) = variable auxiliaire (égale 0 si un mot n'a pas encore reçu de réponse correcte, sinon égale 1)

y = index du mot actuellement demandé

Description du programme:

Lignes 10-60 : Titre

Ligne 70 : Réservation de place en mémoire

Ligne 80 : Affectation de valeurs de départ aux variables

Lignes 90-240 : Menu et branchements dans le programme

Lignes 250-350 : Section de programme "compléter le vocabulaire" On peut soit charger du vocabulaire supplémentaire (saut dans le programme et définition de la variable e), soit entrer du vocabulaire supplémentaire (modification des variables kl et h).

Lignes 360-520 : Section de programme "entrée du vocabulaire"
Vous pouvez entrer jusqu'à 1000 mots. Après l'entrée de 15 mots, un saut est chaque fois effectué au sous-programme "attendre".

Lignes 530-650 : Section de programme "chargement du vocabulaire"
Si le vocabulaire chargé est un vocabulaire supplémentaire, e égale kl+1.

Lignes 660-1370 :

690-770 : Menu

780-910 : Correction, en entrant le mot écrit de façon incorrect et le mot correct. La comparaison de l'entrée "mot incorrect" avec tous les mots stockés dans la mémoire de travail s'effectue dans la boucle de programme des lignes 810-840.

920-1370 : Examen et correction ou suppression des mots.
La sortie des mots (lignes 1040-1080) se fait dans une fenêtre de texte spécialement définie (voir ligne 920). Pour empêcher une fuite vers le haut de la fenêtre de sortie, on affiche jamais que 10 mots à la fois (voir les valeurs initiale et finale de la boucle à la ligne 1040 -première définition en ligne 930- et leur nouvelle définition en ligne 1370).
Les lignes 1090-1360 servent à indiquer le mode d'emploi à l'utilisateur et à corriger les mots. Ce n'est qu'une fois que les mots sortis sont corrects (entrée 0) que d'autre mots à examiner sont sortis (voir ligne 1370).

Sinon, la même sortie se répète (naturellement en tenant compte d'éventuelles corrections, voir ligne 1360).
En lignes 950-1020, lignes qui ne sont atteintes qu'après la sortie de tous les mots (voir ligne 940), on teste si des mots doivent être supprimés (d$(i) est alors égal à " "). Si c'est le cas, les indices des mots suivants sont modifiés en conséquence (voir boucle de programme en lignes 980-1010).
En ligne 1030 se termine cette section de programme, par un retour au menu.

Lignes 1380-1860 : Section de programme "apprendre le vocabulaire"

1410-1420 : Des valeurs de départ sont affectées aux variables

1430-1540 : Explications et menu

1550-1590 : Le nombre aléatoire produit en ligne 1550 (entre 1 et kl) est utilisé kl fois pour déterminer l'index pour l'interrogation sur le vocabulaire (sous-programme en lignes 2100-2260). Si une interrogation correspondante a déjà été effectuée, on prend le nombre aléatoire suivant (voir ligne 1560).

1600-1630 : On détermine les indices des mots qui n'ont pas encore été demandés

1640-1700 : Tant que certains mots reçoivent des réponses incorrectes (voir grandeur auxiliaire hi), les indices correspondants servent de valeurs de départ pour le sous-programme "interrogation"

1710-1860 : Sortie des résultats de l'exercice

Lignes 1870-1960 : Section de programme "sauvegarde du vocabulaire"
Tous les mots présents dans la mémoire de travail sont sauvegardés (comme pour les programmes précédents, il est ici aussi possible d'employer un lecteur de disquette)

Lignes 1970-2090 : Section de programme "fin du programme"
Il est encore possible de sauvegarder ici des mots car on oublie facilement de le faire après un exercice réussi

Lignes 2100-2260 : Sous-programme "interrogation"
Après que l'index y ait été déterminé dans la section de programme "apprendre le vocabulaire", une entrée correspondante est ici demandée. Les réponses justes et fausses sont enregistrées.

Lignes 2270-2340 : Sous-programme "attendre"

8.6 Le puits à idées

- problèmes mathématiques de tous ordres
- équations de réaction et calcul stoechiométrique
- lois de physique
- représentations optiques
- programmation de dictionnaires
- entraînement à l'orthographe
- banque de données de littérature
- analyse de poèmes (voir également le chapitre 8.5)
- stocker et consulter des évènements historiques
- représentation de processus biologiques de croissance
- musique d'après partition
- et bien d'autres choses encore

9. Jouer avec votre CPC

9.1 Remarque préalable

Il y a des gens qui n'utilisent leur ordinateur que pour jouer. Pour eux, le présent ouvrage est bien sûr idéal puisqu'il présente également un certain nombre d'applications "sérieuses" (bien que chacun puisse avoir sa définition du "sérieux"). Néammoins tel ou tel petit jeu peut se révéler très intéressant, surtout si on a soi-même écrit le programme correspondant et qu'on peut ainsi sans problème insérer des variantes supplémentaires dans le déroulement du jeu. Ce qui est dommage, c'est lorsque l'ordinateur est rabaissé au niveau d'un simple instrument de jeu et que ses autres possibilités restent inutilisées. Mais bien entendu, c'est à chacun de savoir ce qu'il veut faire de sa machine.

Ce chapitre vous présente deux programmes de jeu relativement simples qui renferment cependant deux éléments essentiels de la programmation des jeux, la représentation de déplacements et la simulation de "phases de réflexion". Le CPC offre cependant bien d'autres possibilités pour la programmation des jeux électroniques. Vous pourriez par exemple examiner un jour le jeu de caractères complet de votre CPC avec la fonction CHR$(i), en faisant varier la valeur de i de 32 à 255. Peu importe au fond que ce soit un petit homme, un point ou une balle qui parcourt l'écran. Vous êtes donc également invité, dans ce dernier chapitre, à modifier ou compléter les programmes proposés ou à vous en inspirer tout simplement pour réaliser d'autres programmes.

9.2 Tennis vidéo

Il y a dix ans, c'était la grande mode des jeux télé. On pouvait et on peut entre autre jouer ainsi au "tennis". Un tel programme peut être réalisé vraiment facilement.

Le principe consiste à calculer dans le déroulement du programme les coordonnées d'un point qui sera marqué alors que le point précédent sera effacé. Ainsi est créée l'illusion du mouvement. Si le point atteint la limite droite de l'écran, il est relancé si la ligne pouvant être déplacée par le joueur (la raquette) rencontre le point (la balle). Sinon, une nouvelle balle sera jouée.

La raquette est déplacée au moyen des touches curseur. Les "touchés" et "ratés" sont additionnés et affichés. L'utilisateur peut choisir librement la vitesse de la "balle" et de la "raquette" ainsi que la largeur de la "raquette", c'est-à-dire qu'il peut choisir lui-même le niveau de difficulté. Il peut en outre décider si la vitesse de la balle doit varier.

Avec quelques modifications, il est possible d'employer les joysticks pour diriger la raquette. Si vous avez deux joysticks, vous pourriez également introduire dans le jeu une seconde "raquette". Dans ce cas, il vous faudrait toutefois modifier les règles du jeu et l'affichage des points. Vous voyez donc que de nombreuses modifications peuvent être apportées à un tel programme pour qu'il ne devienne jamais ennuyeux. Amusez-vous bien!

Programme:

```
10 REM I24
20 CLS
30 PRINT "I24 - Jeu de balle (tennis a developper)"
40 PRINT:PRINT:PRINT:PRINT:PRINT
50 PRINT TAB(9) "Bernd Kowal, 1985"
60 GOSUB 570
70 PRINT"Entree de parametres :":PRINT:PRINT
80 PRINT"Vitesse de la balle"
90 INPUT"(Valeur entre 10 et 30)        ";b9
100 IF b9<10 OR b9>30 THEN 90
110 PRINT:PRINT"Cette vitesse doit-elle osciller"
120 PRINT"pendant le deroulement"
130 INPUT"du jeu (o/n)                  ";a$
140 IF a$="n" THEN z=0:GOTO 170
150 IF a$="o" THEN z=b9/2:b9=b9*3/4:GOTO 170
160 GOTO 110
```

```
170 PRINT:PRINT"Longueur de la 'raquette'"
180 INPUT"(Valeur entre 10 et 100)     ";sl
190 IF sl<10 OR sl>100 THEN 180
200 PRINT:PRINT"Vitesse de la raquette"
210 INPUT"(Valeur entre  5 et  50)     ";s9
220 IF s9<5 OR s9>50 THEN 210
230 PRINT:PRINT:PRINT"Le jeu est interrompu en appuyant"
240 PRINT"deux fois sur la touche ESC."
250 GOSUB 570
260 KEY DEF 0,1,111
270 KEY DEF 2,1,117
280 SPEED KEY 1,1
290 PLOT 620,0,1:DRAW 0,0,1:DRAW 0,399,1:DRAW 620,399,1
300 a=INT(200-sl/2):b=INT(200+sl/2)
310 PLOT 620,a,1:DRAW 620,b,1
320 GOSUB 510
330 nx=2:ny=200:GOTO 490
340 ax=nx:ay=ny
350 nx=nx+xv:ny=ny+yv
360 IF ny>397 THEN ny=397:yv=-yv
370 IF ny<2 THEN ny=2:yv=-yv
380 IF nx<2 THEN nx=2:xv=-xv
390 ON BREAK GOSUB 630
400 s$=INKEY$
410 IF s$="o" THEN PLOT 620,a,0:a=a+s9:DRAW 620,a,0:b=b+
s9:DRAW 620,b,1
420 IF s$="u" THEN PLOT 620,b,0:b=b-s9:DRAW 620,b,0:a=a-
s9:DRAW 620,a,1
430 IF SGN(xv)=-1 THEN 480
440 IF nx<620 THEN 480
450 nx=620:ny=ay+yv/xv*(620-ax)
460 IF ny>=a AND ny<=b THEN PRINT CHR$(7):t=t+1:GOSUB 51
0:xv=-xv:GOTO 480

470 d=d+1:GOSUB 510:PLOT ax,ay,0:GOTO 330
480 PLOT ax,ay,0
490 PLOT nx,ny,1
500 GOTO 340
510 LOCATE 2,2:PRINT"touches  ";t
520 LOCATE 2,3:PRINT"a cote   ";d
530 xv=ROUND(RND*z+b9)
540 yv=ROUND(RND*20-10)
550 IF yv<5 AND yv>-5 THEN 540
560 RETURN
570 REM Sous-programme attendre
580 LOCATE 5,25
590 PRINT"Frappez une touche S.V.P. !"
600 x$=INKEY$
610 IF x$="" THEN 600
620 CLS:RETURN
630 SPEED KEY 10,4:END
```

Liste de variables:

a = limite inférieure de la coordonnée y pour la raquette

a$ = chaîne de réponse (o/n)

ax = "ancienne" coordonnée x de la balle

ay = "ancienne" coordonnée y de la balle

b = limite supérieure de la coordonnée y pour la raquette

bg = vitesse de la balle

d = nombre de balles manquées

nx = "nouvelle" coordonnée x de la balle

ny = "nouvelle" coordonnée y de la balle

s$ = valeur de la touche appuyée pour la commande de la raquette

sg = vitesse de la raquette

sl = longueur de la raquette

t = nombre de balles touchées

xv = vitesse de déplacement de la balle dans la direction x

yv = vitesse de déplacement de la balle dans la direction y

z = variable auxiliaire (égale bg/2 si la vitesse de la balle, sinon égale 0)

Description du programme:

Lignes 10-60 : Titre

Lignes 70-250 : Entrée des paramètres et sortie d'une explicaton du mode d'emploi du programme. Les limites imposées pour les valeurs demandées ont été choisies simplement en fonction du critère d'un bon déroulement du jeu. Les calculs en ligne 150 servent à déterminer la vitesse de déplacement variant de façon aléatoire en ligne 530.

Lignes 260-270 : Les touches de commande du curseur se voient affecter des valeurs. Lorsqu'on appuie sur la touche "curseur haut", un "h" (pour haut) sera renvoyé et un "b" (comme bas) pour la touche "curseur bas".

Ligne 280 : De nouvelles valeurs sont fixées pour la vitesse de répétition d'une touche restant enfoncée, de façon à augmenter la mobilité de la raquette

Ligne 290 : Dessin du cadre de jeu

Ligne 300 : Calcul des coordonnées limites (direction y) pour la raquette

Ligne 310 : Dessin de la raquette

Ligne 320 : Saut au sous-programme (ligne 510) pour déterminer la vitesse de déplacement de la balle

Ligne 330 : Les coordonnées de départ de la balle sont calculées et la balle est dessinée directement (saut à la ligne 490)

Lignes 340-350 : La nouvelle position de la balle est déterminée (ancienne position + vitesse de déplacement), après que l'ancienne position ait été placée dans les tableaux ax et ay

Lignes 360-380 : Si lors du changement de position la limite supérieure, inférieure ou gauche est atteinte ou dépassée, on prend comme coordonnée maximale une valeur directement à côté du bord et la direction du déplacement est inversée (yv = -yv ou xv = -xv)

Ligne 390 : Après que la touche ESC ait été appuyée par deux fois, on saute au sous-programme (ligne 630) qui termine le jeu et le programme

Lignes 400-420 : La valeur de la touche enfoncée est renvoyée. Le fait d'appuyer sur la touche "curseur haut" a pour effet de faire tracer une ligne "sans couleur" entre la position la plus basse de la raquette et le bas de la nouvelle position de la raquette (a=a+sg), en tenant bien sûr compte de la vitesse indiquée pour la raquette. Une ligne (colorée) est également tracée entre le bas et le haut (b=b+sg) de la nouvelle position de la raquette. On procède de même pour la touche "curseur bas".

Ligne 430 : Si le point se déplace de la droite vers la gauche (c'est donc que la balle est renvoyée), le point peut être dessiné directement (saut à la ligne 480)

Lignes 440-470 : On teste si la balle a rencontré la raquette. Le problème ne se pose pas si la limite droite de l'écran n'a pas encore été atteinte (ligne 440). Si elle est atteinte, les coordonnées nx et ny sont fixées en ligne 450 de façon à ce qu'elles figurent sur la ligne allant de la raquette à la coordonnée x 620 (voyez le schéma à la suite de cette description).

Si la coordonnée ny se trouve alors à l'intérieur de l'intervalle de limites a et b, une "touche" a été réussie (bip, t=t+1). On détermine alors dans le sous-programme (ligne 510) de nouvelles vitesses de déplacement et la balle est renvoyée (xv = -xv). Si ny ne figure pas à l'intérieur de l'intervalle (manqué), une nouvelle balle est lancée, avec de nouvelles vitesses de déplacement (à partir de la ligne 330), après qu'un point sans couleur ait été dessiné dans l'ancienne position.

Ligne 480 : Suppression de "l'ancien" point

Ligne 490 : Dessin du "nouveau" point

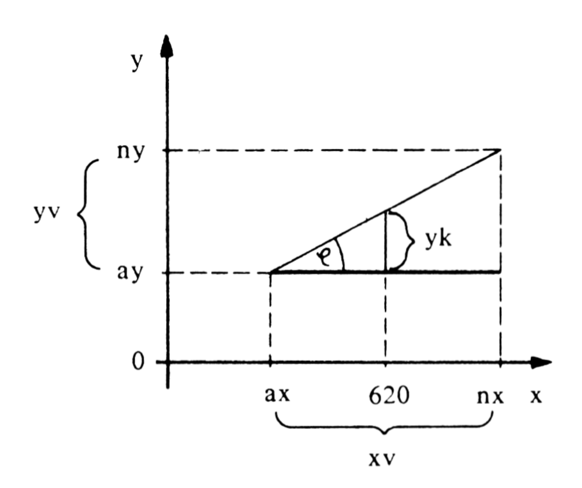
Ligne 500 : Retour à la ligne 340 pour continuer le jeu

Lignes 510-560 : Sous-programme de sortie du taux de réussite et de détermination des vitesses de déplacement de la balle. En ligne 530, les valeurs pour xv peuvent varier de 25% en plus ou en moins de la vitesse de balle indiquée, si cela a été voulu ainsi (lignes 110-150). Pour yv, les valeurs sont tirées au hasard. Elles sont comprises entre -10 et -5 ou entre 5 et 10.

Lignes 570-620 : Sous-programme "attendre"

Ligne 630 : La vitesse de répétition d'une touche maintenue appuyée revient à la "normale" et le programme est terminé.

Schéma décrivant comment la coordonnée ny est déterminée en ligne 450:



recherché: yk

$$\tan \varphi = \frac{yv}{xv} \quad \text{et} \quad \tan \varphi = \frac{yk}{620-ax}$$

donc:

$$\frac{yv}{xv} = \frac{yk}{620-ax}$$

$$yk = \frac{yv}{xv} * (620-ax)$$

on obtient pour ny:

ny = ay + yk = ay + yv / xv * (20 - ax)

9.3 Puissance quatre

Que faites-vous si vous voulez jouer à un jeu pour deux personnes? La réponse est simple: vous cherchez un partenaire de jeu. Mais que faites-vous si personne ne veut jouer avec vous? Il n'y de solution simple à cette question que pour ceux qui possèdent un ordinateur: vous chargez un programme et l'ordinateur remplacera le partenaire de jeu manquant.

Mais comment peut-on arriver à ce que l'ordinateur "réfléchisse", comme il convient pour un partenaire de jeu? Il vous faut d'abord lui expliquer le terrain de jeu et les règles du jeu. Dans "Puissance quatre", le terrain se compose d'une grille de 8 cases sur 6. Le but du jeu est de constituer une ligne de quatre jetons (horizontale, verticale ou en diagonale). Vos jetons doivent bien sûr être distingués de ceux de votre partenaire de jeu.

Pour obtenir une force moyenne de jeu, il suffit de faire effectuer par votre ordinateur les réflexions suivantes:

1) Puis-je (moi, l'ordinateur) gagner en un coup?

2) Puis-je (moi, l'ordinateur) au coup suivant empêcher que mon adversaire gagne la partie?

Il est en outre possible d'intégrer des variantes de tactique de jeu dans le déroulement du programme. Pour "Puissance quatre", une tactique, qui mène à la victoire, pourrait consister à essayer de constituer une ligne de trois jetons qui ne soit limitée d'aucun côté par des jetons adverses. Dans les différentes sections du programme, il faudrait alors intégrer des instructions supplémentaires qui commenceraient par

```
IF h=2 AND ... THEN
```

Dans le programme suivant, nous nous limitons toutefois aux deux premières étapes de réflexion indiquées. Si aucune des deux propositions n'est vérifiée, l'ordinateur ne se livre pas à une réflexion tactique mais fait jouer le générateur de hasard. On ne tient donc pas compte des effets du coup qui viendra après le prochain coup.

Un mot concernant la technique de programmation:

L'ordinateur réfléchit en couvrant toute la surface de jeu, c'est-à-dire qu'il peut arriver que soient examinés un très grand nombre de cases qui n'ont encore aucune importance pour le jeu. Une autre technique de programmation, qui réduirait considérablement les temps de calcul dans la phase initiale du jeu, consisterait à limiter la réflexion aux cases occupées. Les différentes instructions IF ... THEN resteraient inchangées dans leur principe.

Il ne nous reste plus qu'à vous souhaiter beaucoup de réussite dans le combat homme contre machine. Faites attention: votre ordinateur ne manquera jamais de remarquer si trois jetons figurent sur une ligne...

Programme:

```
10 REM I25
20 CLS
30 PRINT TAB(9)"I25 - Puissance quatre"
40 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
50 PRINT TAB(10) "Bernd Kowal, 1985"
60 GOSUB 1180
70 PRINT"Regles du jeu :":PRINT:PRINT
80 PRINT"Le but du jeu est de placer quatre"
90 PRINT"jetons sur une ligne (peu importe que"
100 PRINT"ce soit horizontalement, verticalement"
105 PRINT"ou en diagonale)."
110 PRINT"Le terrain de jeu est une grille de 8"
120 PRINT"cases sur 6. Les lignes sont numerotees"
130 PRINT"de 1 a 8. Si vous voulez placer par"
140 PRINT"exemple un jeton sur la deuxieme ligne,"
150 PRINT"vous devez entrer un 2. Le jeton"
160 PRINT"tombera jusqu'a ce qu'il tombe sur un"
170 PRINT"autre jeton ou dans la derniere ligne."
180 PRINT
190 PRINT"Voici mes jetons     : ";CHR$(143);CHR$(143);
200 PRINT TAB(27) CHR$(143);CHR$(143);CHR$(143):PRINT
210 PRINT"et voici vos jetons : ";CHR$(207);CHR$(207);CH
R$(207)
220 PRINT TAB(27) CHR$(207);CHR$(207);CHR$(207):PRINT
230 PRINT:INPUT"Voulez-vous commencer (o/n) ";a$
240 IF a$<>"n" AND a$<>"o" THEN 230
250 CLS:FOR i=1 TO 7:PRINT"  ";
260 IF i=7 THEN PRINT CHR$(147);:GOTO 280
270 PRINT CHR$(151);
280 FOR j=1 TO 31:PRINT CHR$(154);:NEXT j
290 IF i=7 THEN PRINT CHR$(153):GOTO 310
300 PRINT CHR$(157)
310 PRINT:PRINT:NEXT i
320 j=3
330 FOR i=2 TO 19 STEP 3
340 LOCATE j,i:PRINT CHR$(149)
350 LOCATE j,i+1:PRINT CHR$(149)
360 NEXT i
370 IF j=3 THEN j=35:GOTO 330
380 FOR j=7 TO 31 STEP 4
390 FOR i=1 TO 16 STEP 3
400 LOCATE j,i:PRINT CHR$(159)
410 LOCATE j,i+1:PRINT CHR$(149)
420 LOCATE j,i+2:PRINT CHR$(149)
430 NEXT i:LOCATE j,i:PRINT CHR$(155):NEXT j
440 PRINT"    ";:FOR i=1 TO 8
450 PRINT i;" ";
```

```
460 r(i,0)=1:NEXT i
470 IF a$="n" THEN s=ROUND(RND*4+2.5):cd=143:GOSUB 1080:
co(s,j)=1:GOSUB 1110
480 LOCATE 3,23:INPUT"Votre choix ";s
490 IF s<1 OR s>8 THEN 480
500 cd=207:GOSUB 1080:sb(s,j)=1:GOSUB 1110
510 cd=143:FOR i=1 TO 8
520 FOR j=6 TO 3 STEP -1
530 IF r(i,j)=0 THEN 600
540 IF p=1 THEN 580
550 IF co(i,j)=0 THEN 610
560 IF co(i,j-1)=1 AND co(i,j-2)=1 THEN s=i:GOSUB 1080:G
OSUB 1110:GOTO 1060
570 GOTO 600
580 IF sb(i,j)=0 THEN 610
590 IF sb(i,j-1)=1 AND sb(i,j-2)=1 THEN r(s,j1)=0:co(s,j
1)=0:GOTO 1000
600 NEXT j
610 NEXT i
620 FOR j=1 TO 6
630 sb1=1:sb2=4:h=0
640 FOR i=sb1 TO sb2
650 IF p=1 THEN 680
660 IF co(i,j)=0 THEN i1=i ELSE h=h+1
670 GOTO 690
680 IF sb(i,j)=0 THEN i1=i ELSE h=h+1
690 NEXT i
700 IF p=0 AND h=3 AND r(i1,j)=0 AND r(i1,j-1)=1 THEN s=
i1:GOSUB 1080:GOSUB 1110:GOTO 1060
710 IF p=1 AND h=3 AND r(i1,j)=0 AND r(i1,j-1)=1 THEN r(
s,j1)=0:co(s,j1)=0:GOTO 1000
720 IF sb1<6 THEN h=0:sb1=sb1+1:sb2=sb2+1:GOTO 640
730 NEXT j
740 FOR i=1 TO 5
750 FOR j=1 TO 3
760 ip=i:jp=j:h=0
770 IF p=1 THEN 800
780 IF co(ip,jp)=0 THEN ip1=ip:jp1=jp ELSE h=h+1
790 GOTO 810
800 IF sb(ip,jp)=0 THEN ip1=ip:jp1=jp ELSE h=h+1
810 IF ip<i+3 THEN ip=ip+1:jp=jp+1:GOTO 770
820 IF p=0 AND h=3 AND r(ip1,jp1)=0 AND r(ip1,jp1-1)=1 T
HEN s=ip1:GOSUB 1080:GOSUB 1110:GOTO 1060
830 IF p=1 AND h=3 AND r(ip1,jp1)=0 AND r(ip1,jp1-1)=1 T
HEN r(s,j1)=0:co(s,j1)=0:GOTO 1000
840 NEXT j
850 NEXT i
860 FOR i=4 TO 8
```

```
870 FOR j=1 TO 3
880 ip=i:jp=j:h=0
890 IF p=1 THEN 920
900 IF co(ip,jp)=0 THEN ip1=ip:jp1=jp ELSE h=h+1
910 GOTO 930
920 IF sb(ip,jp)=0 THEN ip1=ip:jp1=jp ELSE h=h+1
930 IF ip>i-3 THEN ip=ip-1:jp=jp+1:GOTO 890
940 IF p=0 AND h=3 AND r(ip1,jp1)=0 AND r(ip1,jp1-1)=1 T
HEN s=ip1:GOSUB 1080:GOSUB 1110:GOTO 1060
950 IF p=1 AND h=3 AND r(ip1,jp1)=0 AND r(ip1,jp1-1)=1 T
HEN r(s,j1)=0:co(s,j1)=0:GOTO 1000
960 NEXT j
970 NEXT i
980 IF p=1 THEN p=0:GOSUB 1110:GOTO 480
990 FOR i=1 TO 8:s(i)=0:NEXT i:l=0
1000 l=l+1:IF l>8 THEN 1050
1010 IF l=1 THEN s(l)=ROUND(RND*8+0.5):GOTO 1030
1020 s(l)=s(l-1)+1:IF s(l)=9 THEN s(l)=1
1030 s=s(l):GOSUB 1080:co(s,j)=1:j1=j
1040 p=1:GOTO 510
1050 LOCATE 19,23:PRINT"Vous avez gagne":GOTO 1070
1060 LOCATE 19,23:PRINT"J'ai gagne"
1070 END
1080 FOR j=1 TO 6
1090 IF r(s,j)=0 THEN r(s,j)=1:RETURN
1100 NEXT j:IF cd=207 THEN 480 ELSE 1000
1110 z=7:FOR i=2 TO 17 STEP 3
1120 z=z-1:k=0
1130 IF r(s,z)=1 AND r(s,z+1)=1 THEN RETURN
1140 LOCATE s*3+s,i:PRINT CHR$(cd);CHR$(cd);CHR$(cd)
1150 LOCATE s*3+s,i+1:PRINT CHR$(cd);CHR$(cd);CHR$(cd)
1160 IF r(s,z)=0 AND k=0 THEN PEN 0:k=1:GOTO 1140
1170 PEN 1:NEXT i:RETURN
1180 REM SP attendre
1190 LOCATE 7,25
1200 PRINT"Frappez une touche S.V.P. !"
1210 x$=INKEY$
1220 IF x$="" THEN 1210
1230 CLS:RETURN
```

Liste de variables:

a$ = chaîne de caractères de réponse (o/n)

cd = dessin du jeton, expression numérique pour le dessin des jetons dans le sous-programme en lignes 1110-1170

co(i,j) = tableau de la grille (i pour la colonne et j pour la ligne); 1 pour occupé et 0 pour libre

h = variable de comptage pour le nombre de jetons sur une ligne

i = index de comptage

il = index de colonne pour une colonne non occupée

ip = index de colonne dans l'examen des diagonales

ip1 = index de colonne dans l'examen des diagonales pour une case non occupée

j = index de comptage

jp = index de ligne dans l'examen des diagonales

jp1 = index de ligne dans l'examen des diagonales pour une case non occupée

k = variable auxiliaire dans le sous-programme en lignes 1110-1170

l = index de comptage pour la valeur de la colonne à tester

p = variable auxiliaire qui indique laquelle des étapes de réflexion est en train d'être exécutée

r(i,j) = tableau de la grille (i pour la colonne et j pour la ligne); 1 pour occupé et 0 pour libre

s,s(i) = valeur de colonne choisie ou calculée ou déterminée au hasard

sb(i,j) = tableau de la grille (i pour la colonne et j pour la ligne); 1 pour occupé et 0 pour libre

sb1 = limite inférieure d'une boucle de programme

sb2 = limite supérieure d'une boucle de programme

z = index de ligne dans le sous-programme en lignes 1110-1170

Description du programme:

Lignes 10-60 : Titre

Lignes 70-240 : Sortie des règles du jeu et questions concernant le début du jeu

Lignes 250-460 : Le terrain de jeu est dessiné. Les cases de la grille de ligne 0 sont en outre fixées sur 1 pour indiquer à l'ordinateur qu'un jeton ne peut être placé plus bas que la première ligne.

Ligne 470 : Un nombre aléatoire entre 3 et 6 est formé si l'ordinateur doit commencer le jeu. Les cases correspondantes sont en outre occupées et le jeton est dessiné.

Lignes 480-500 : Choix de la colonne par l'utilisateur du programme, occupation de la case et dessin du jeton

Lignes 510-1040 : L'ordinateur réfléchit

Cette section de programme est d'abord parcourue avec p=0, c'est-à-dire que l'ordinateur examine s'il peut mettre fin au jeu en gagnant. Si ce n'est pas le cas, un parcours est effectué avec p=1, c'est-à-dire que l'ordinateur teste si le nombre aléatoire formé en ligne 1010 ne pourrait pas avoir pour conséquence de faire gagner l'adversaire. Ce nombre aléatoire n'est utilisé par l'ordinateur pour jouer (ligne 980) que si pendant le parcours on n'a pas sauté à la ligne 1000 (test d'un nouveau nombre). On effectue toujours un saut à la ligne 1000 lorsque le choix d'un nombre ou d'une colonne entraînerait la défaite de l'ordinateur. Un tel saut entraîne en même temps une remise à zéro des cases r(s,j1) et co(s,j1) occupées en ligne 1030 pour effectuer les tests. Au maximum, les 8 colonnes possibles seront essayées (lignes 1000-1020, seul le premier nombre est tiré au hasard). Après 8 parcours sans succès, l'ordinateur abandonne et saute à la fin du programme (avec sortie d'un commentaire en ligne 1050), c'est-à-dire que l'utilisateur du programme devra parfois se demander pourquoi il a gagné.

On peut distinguer les sections de programme suivantes:

510-610 : Examen des verticales
620-730 : Examen des horizontales
740-850 : Examen des diagonales allant d'en bas à gauche à en haut à droite
860-970 : Examen des diagonales allant d'en bas à droite à en haut à gauche

980-1040 : Dessin d'un jeton et saut au choix de l'adversaire si un parcours a été effectué avec succès; formation de nombres de test, occupation-test des cases r(s,j1) et co(s,j1) et retour à la ligne 510

Lignes 1050-1060 : Commentaire sur le résultat du jeu

Ligne 1070 : Fin du programme

Lignes 1080-1100 : Sous-programme pour occuper la case r(s,j) en fonction de la colonne s sélectionnée. Si 6 jetons figurent déjà dans la colonne, un retour est effectué pour un nouveau choix (la variable cd indiquant si c'est l'ordinateur ou l'utilisateur du jeu qui a fait ce choix).

Lignes 1110-1170 : Sous-programme de dessin d'un jeton

Lignes 1180-1230 : Sous-programme "attendre"

9.4 Le puits à idées

- Jeux de damiers, comme les dames, la marelle ou les échecs
- Jeux de cartes comme la belote, les tarots, etc...
- Jeux de dés
- Scrabble (marque déposée)
- Monopoly (marque déposée) et autres jeux de stratégie
- Jeux de labyrinthes
- Bataille navale
- Alunissage
- Jeux d'envahisseurs de toutes sortes
- et bien d'autres choses encore

9.5 La chenille :

Présentation :

Pour conclure cette série de programmes basics, voici le listing d'un jeu en langage machine.

Ce langage qui effraie les débutants est quand même accessible à toutes les personnes qui maîtrisent le basic.

Nous espèrons que cet exemple de programmation ludique vous aidera à pénétrer dans le monde du langage machine.

Programme :

```
;          I 26    :    CHENILLE.
;
;  (C) 1985 MICRO APPLICATION -PS-.
;
;
         ORG  38000
;
         ENT  DEP
BREAK    EQU  5000                          ;Adresse a laquelle est chargee DAMS
                                                ou adresse d'un moniteur.
;
;
; Definition des vecteurs utilises.
;
SETMODE  EQU  #BC0E                         ;Fixe mode.
WINDEN   EQU  #BB66                         ;Cree fenetre.
WINDOW   EQU  #BBB4                         ;Selectionne fenetre.
CLW      EQU  #BB6C                         ;Efface fenetre.
PRINT    EQU  #BB5A                         ;Affiche caractere.
RDCHAR   EQU  #BB60                         ;Lis caractere a l'ecran.
SETROW   EQU  #BB72                         ;Fixe numero de ligne.
SETCOL   EQU  #BB6F                         ;Fixe numero de colonne.
SETPEN   EQU  #BB90                         ;Fixe stylot.
SETINK   EQU  #BC32                         ;Fixe encre.
KMRESET  EQU  #BB03                         ;Reset clavier.
WAITCHAR EQU  #BB06                         ;Attend un caractere.
READCHAR EQU  #BB09                         ;Lis clavier.
TESTKEY  EQU  #BB1E                         ;Teste touche.
SDRESET  EQU  #BCA7                         ;Reset son.
SDAMPLI  EQU  #BCBC                         ;Cre une enveloppe.
SDQUEUE  EQU  #BCAA                         ;Ajoute un son.
SDRELEAS EQU  #BCB3                         ;Produit un son.
;
; Definition des constantes.
;
GROS     EQU  3                             ;Grossissement de digestion.
LONGUEUR EQU  256                           ;Taille de la chenille.
;
INK0     EQU  0                             ;Encre 0: noire.
INK1     EQU  23                            ;Encre 1: turquoise pastel.
INK2     EQU  6                             ;Encre 2: Rouge vif.
INK3     EQU  18                            ;Encre 3: vert vif.
;
; Definition des variables.
;
AX
NIVEAU   DEFB                               ;Niveau de jeux.
TETE     DEFW                               ;Adresse de la tete.
QUEUE    DEFW                               ;Adresse de la queue.
DIGERE   DEFB                               ;Duree de la digestion.
DIRECT   DEFB                               ;Direction.
SEED     DEFW                               ;Nombre aleatoire.
SCORE    DEFW
HISCORE  DEFS 2
VITESSE  DEFB 64
NBFRUIT  DEFB                               ;Nombre de fruits.
COORD    DEFW                               ;Coordonnees de la tete.
;
;
DEP
         CALL INITP
         CALL INITJEU                       ;Score = 0 , Niveau = 1.
         CALL AFWIND1                       ;Affiche la fenetre 1.
         CALL HELP                          ;Affiche les instructions.
```

```
;
;
JEUL0
        CALL INITJEU                    ;Score = 0 , Niveau = 1.
JEUL1
        CALL AFWIND1                    ;Affiche la fenetre 1.
        CALL AFTAB                      ;Affiche le tableau de jeux.
        CALL READY                      ;Joueur Pret ?
        LD   A,2                        ;Active la fenetre 2.
        CALL WINDOW
        LD   HL,BUFFER                  ;Adresse de debut  de la chenille.
        LD   (QUEUE),HL                 ;Fixe l'adresse de la queue.
        LD   BC,10*256+18               ;B = ligne 10, C = colonne 18.
        LD   (HL),B                     ;Fixe point de
        INC  HL                         ;  depart de
        LD   (HL),C                     ;  la chenille.
        LD   (COORD),BC
        INC  HL
        LD   (TETE),HL                  ;Fixe l'adresse de la tete.
        CALL LOCATE                     ;Positionne le curseur.
        LD   A,32
        CALL PRINT                      ;Efface ce point.
        LD   A,255                      ;Signal: reste immobile.
        LD   (DIRECT),A
        LD   (IX+DIGERE-AX),0           ;Initialise la digestion.

        CALL JELO                       ;Lance le jeux.
        JR   NC,CRASH                   ;Si la chenille est morte, alors fin.
        INC  (IX+NIVEAU-AX)             ;Sinon passe au niveau superieur.
        JR   JEUL1                      ;Continue le jeux.
;
CRASH
        LD   A,MORT                     ;Caractere de mort.
        LD   DE,20                      ;Frequence du son.
        CALL EFCHEN                     ;Efface la Chenille.
        LD   A," "                      ;Caractere d'effacement.
        LD   DE,100                     ;Frequence du son.
        CALL EFCHEN                     ;Efface la chenille.
        LD   DE,(SCORE)
        LD   HL,(HISCORE)
        AND  A
        SBC  HL,DE                      ;Score > hiscore ?
        JP   NC,JEUL0                   ;Non alors debut de jeux.
        LD   (HISCORE),DE               ;Oui alors stoke le nouveau.
        JP   JEUL0                      ;Debut de jeux.
;
;
JELO
;
; Boucle principale du jeux.
;
; Entree: pas de registres.
; Sortie: Carry a 1 => le tableau est fini.
;       : Carry a 0 => la chenille est morte.
;
        CALL FTDIR                      ;Teste clavier, modifie la direction.
        CALL CALPOS                     ;Calcule nouvelle position de la tete
        LD   (COORD),BC                 ;  et stocke la.
        CALL TESTPOS                    ;Teste si la place est occupee.
        PUSH AF                         ;Sauvegarde le flag.
        CALL Z,MANGE                    ;Si c'est un fruit alors mange le.
        CALL LOCATE                     ;Positionne le curseur
        LD   A,240                      ;  et affiche la tete.
        CALL PRINT
```

```
        CALL CHAINE                     ;Enchaine la tete et efface la queue.
        POP  AF                         ;Retire le flag de collision.
        RET  NC                         ;Si il y a eu collision alors fin.
        LD   A,(NBFRUIT)
        AND  A                          ;Reste t-il des fruits ?
        JR   NZ,JELO                    ;Si oui alors continue.
        SCF                             ;Signal: Tableau fini.
        RET
;
INITJEU
        LD   A,1
        LD   (NIVEAU),A
        LD   HL,0
        LD   (SCORE),HL
        RET
;
;
EFCHEN
;
; Routine d'effacement de la chenille.
;
; Entree: A  = caractere d'effacement.
;         DE = hauteur initiale du son.
;
; Sortie: pas de registres.
;
        LD   HL,(TETE)                  ;HL = adresse de la tete.
EFCHENL1
        PUSH AF                         ;Sauvegarde le caractere d'effacement.
        LD   BC,BUFFER+1                ;Bas du buffer de la chenille.
        AND  A                          ;Met carry a 0.
        SBC  HL,BC                      ;La tete est au debut du buffer ?
        ADD  HL,BC                      ;Restaure HL sans perturber carry.
        JR   NC,EFCHENP1                ;Si non alors passe.
        LD   HL,ENDBUFF                 ;Si oui alors met la tete en haut.
EFCHENP1 DEC HL
        LD   C,(HL)                     ;Met coord. colonne de la tete dans C.
        DEC  HL
        LD   B,(HL)                     ;Met coord. ligne de la tete dans B.
        PUSH AF                         ;Sauvegarde caractere d'effacement.
        CALL LOCATE                     ;Posisionne le curseur.
        POP  AF                         ;Retire le caractere d'effacement
        CALL PRINT                      ;  et efface.
        PUSH HL                         ;Sauvegarde adresse de la tete.
        PUSH DE                         ;Sauvegarde frequence du son.
        LD   HL,0                       ;Signal: duree est definie par envelop.
        LD   A,1                        ;Signal: enveloppe numero 1.
        CALL SON                        ;Produit le son.
        POP  DE                         ;Retire frequence du son.
        LD   HL,30
        ADD  HL,DE                      ;Baisse la frequece.
        EX   DE,HL                      ;Remet le resultat dans DE.
        LD   A,20                       ;Attend 20 unitees.
        CALL TEMPO
        POP  HL                         ;Retire adresse de la tete.
        POP  AF                         ;Retire caractere d'effacement.
        AND  A                          ;Met carry a 0.
        LD   BC,(QUEUE)                 ;BC prends l'adresse de la queue.
        PUSH HL
        SBC  HL,BC                      ;Toute la chenille a ete effacee ?
        POP  HL
        JR   NZ,EFCHENL1                ;Si non alors continue.
        RET                             ;Si oui alors fin.
;
```

```
;
TEMPO
;
; Produit une temporisation dont la duree depend de A.
;
          LD   B,A
TEMPOL2
          LD   C,0
TEMPOL1
          DEC  C
          JR   NZ,TEMPOL1
          DJNZ TEMPOL2
          RET
;
;
;
FTDIR
;
; Teste le clavier et calcule nouvelle direction.
;
; Entree: Pas de registres.
;
; Sortie: A  = code de direction.
;
          LD   D,(IX+DIRECT-AX)        ;D prends l'ancienne direction.
          CALL LICLAV                  ;Lit le clavier.
          LD   A,D                     ;Passe le code de direction a A.
          RET  NC                      ;Revient si direction n'a pas change.
          AND  3                       ;Modulo 4  (pour les 4 directions).
          LD   (DIRECT),A              ;Stocke nouvelle direction.
          RET
;
LICLAV
          LD   A,66                    ;Teste la touche Escape.
          CALL TESTKEY
          JP   NZ,BREAK                ;Si elle est enfoncee alors exit.
          CALL READCHAR                ;Retire catactere du buffer clavier.
          RET  NC                      ;Revient si il n'y en avait pas.
          CP   "/"                     ;Touche "a gauche" ?
          JR   NZ,LICLAVP3             ;Si non alors passe.
          DEC  D                       ;Calcule nouvelle direction a gauche.
          SCF                          ;Signal: la direction a change.
          RET
LICLAVP3
          CP   "\"                     ;Touche "a droite" ?
          JR   NZ,LICLAVP4             ;Si non alors passe.
          INC  D                       ;Calcule nouvelle direction a droite.
          SCF                          ;Signal: la direction a change.
          RET
LICLAVP4
          OR   A                       ;Signal: la direction n'a pas change.
          RET
;
CALPOS
;
; Calcule la nouvelle position de la tete d'apres la direction.
;
; Entree: A  = code de direction.
;
; Sortie: BC = coordonnees de la tete.
;
          LD   BC,(COORD)              ;Ancienne coordonnees de la tete.
          AND  A                       ;A = 0 ?
          JR   Z,DIRDROI               ;Si oui alors direction a droite.
```

```
        DEC  A                          ;A = 1 ?
        JR   Z,DIRBAS                   ;Si oui alors direction vers le bas.
        DEC  A                          ;A = 2 ?
        JR   Z,DIRGAU                   ;Si oui alors direction a gauche.
        DEC  A                          ;A = 3 ?
        RET  NZ                         ;Retour si direc. pas vers le haut.
;
        DEC  B                          ;Passe a la ligne superieure.
        RET
DIRBAS
        INC  B                          ;Passe a la ligne de dessous.
        RET
DIRDROI
        INC  C                          ;Passe a la colonne de droite.
        RET
DIRGAU
        DEC  C                          ;Passe a la colonne de gauche.
        RET
;
;
TESTPOS
;
; Teste si la position de la tete est un espace.
;
; Entree: BC = coordonnees ligne et colonne de la tete.
;
; Sortie: carry = 0  => la chenille a heurte un obstacle.
;         carry = 1  => OK.
;             Z = 1  => la chenille a mange un fruit.
;             Z = 0  => OK.
;
        CALL LOCATE                     ;Positionne le curseur.
        CALL RDCHAR                     ;Lis le caractere a l'ecran.
        CCF
        RET  C                          ;Retour avec signal OK si illisible.
        CP   FRUIT                      ;Est-ce un fruit ?
        SCF                             ;Signal: OK.
        RET  Z                          ;Retour si oui.
        CP   33                         ;Si ce n'est pas un espace
        RET                             ;  alors retour avec carry annule.
;
MANGE
;
; Produit un son, calcule et affiche le score.
;
        PUSH BC                         ;Sauvegarde les coordonnees de la tete.
        LD   A,(NIVEAU)
        LD   E,A
        ADD  A,A                        ;Multiplie par 2.
        ADD  A,A                        ;Par 4.
        ADD  A,E                        ;Par 5.
        ADD  A,A                        ;Par 10.
        LD   E,A                        ;Passe le resultat a E.
        LD   A,(VITESSE)
        RRCA                            ;Divise par 2.
        RRCA                            ;Par 4.
        RRCA                            ;Par 8.
        LD   D,A                        ;Passe le resultat a D.
        LD   A,31
        SUB  D                          ;Inverse l'intervalle 0-31.
        RLCA                            ;Multipli par 2.
        ADD  A,E                        ;Ajoute le premier total.
        LD   E,A                        ;Passe le resultat a E.
        LD   D,0                        ;Reset le hight-byte.
```

```
        LD    HL,(SCORE)
        ADD   HL,DE                     ;Met le score a jour
        LD    (SCORE),HL                ;  et sauvegarde le.
        EX    DE,HL
        LD    A,1                       ;Active la fenetre 1.
        CALL  WINDOW
        LD    BC,256+4                  ;Ligne 1, Colone 4.
        CALL  LOCATE                    ;Positionne le curseur.
        CALL  OUTNB                     ;Affiche le score.
        LD    HL,0                      ;Signal: utilise l'enveloppe.
        LD    DE,700                    ;Frequence du son.
        LD    A,1                       ;Signal: Enveloppe numero 1.
        CALL  SON                       ;Produit le son.
        LD    A,2                       ;Active la fenetre 2.
        CALL  WINDOW
        LD    A,(DIGERE)                ;A = etat de digestion.
        ADD   A,GROS                    ;Ajoute la digestion actuelle
        LD    (DIGERE),A                ;  et stocke la.
        DEC   (IX+NBFRUIT-AX)           ;Decremente le nombre de fruits.
        POP   BC                        ;Retire coordonnees de la tete.
        RET
;
;
CHAINE
;
; Ajoute la derniere position de la tete dans la chaine.
;
; Entree: BC = Coordonnees ligne et colonne de la tete.
;
        LD    HL,(TETE)                 ;Adresse de la tete.
        LD    (HL),B                    ;Ajoute nouvelle ligne.
        INC   HL
        LD    (HL),C                    ;Ajoute nouvelle colonne.
        INC   HL
        CALL  TESTENDB                  ;Teste si la fin du buffer est atteinte
        LD    (TETE),HL                 ;Stocke la nouvelle adresse de la tete.
        LD    A,(DIGERE)
        AND   A                         ;Cycle de digestion ?
        JR    Z,EFFQUEUE                ;Si non passe.
        CP    GROS                      ;Est-ce le premier cycle ?
        LD    A,(VITESSE)
        CALL  NZ,TEMPO                  ;Si non ajoute temporisation.
        DEC   (IX+DIGERE-AX)            ;Un cycle de digestion en moins.
        RET
EFFQUEUE
        LD    A,(VITESSE)
        CALL  TEMPO                     ;Attend.
        LD    HL,(QUEUE)                ;Adresse de la queue.
        LD    B,(HL)                    ;Coord. ligne de la queue.
        INC   HL
        LD    C,(HL)                    ;Coord. colonne de la queue.
        INC   HL
        CALL  TESTENDB                  ;Teste si fin du buffer est atteinte.
        LD    (QUEUE),HL                ;Stocke nouvelle adresse de la queue.
        CALL  LOCATE                    ;Positionne le curseur,
        LD    A,32                      ;  et efface la queue.
        JP    PRINT
;
TESTENDB
;
; Teste si fin du buffer est atteinte.
;
        LD    DE,ENDBUFF
        AND   A
```

```
        SBC   HL,DE
        ADD   HL,DE
        RET   C
        LD    HL,BUFFER
        RET
;
;
AFWIND1
;
; Affiche la Premiere fenetre.
;
        LD    A,1                 ;Active la fenetre 1.
        CALL  WINDOW
        LD    BC,256+1            ;Ligne 1, colonne 1.
        CALL  LOCATE              ;Positionne le curseur.
        CALL  PRTL                ;Affiche chaine de caracteres.
        DEFM  SC:
        DEFB  -1                  ;Signal: fin de chaine.
        LD    DE,(SCORE)
        CALL  OUTNB               ;Affiche le score.
        LD    BC,256+10           ;Ligne 1, colonne 10.
        CALL  LOCATE              ;Positionne le curseur.
        CALL  PRTL                ;Affiche chaine de caracteres qui suit.
        DEFM  NIVEAU:
        DEFB  -1                  ;Signal: fin de chaine.
        LD    A,(NIVEAU)
        ADD   A,48                ;Transforme le niveau en code ASCII,
        CALL  PRINT               ;  et affiche le.
        LD    BC,256+20           ;Ligne 1, colonne 20.
        CALL  LOCATE              ;Positionne le curseur.
        CALL  PRTL                ;Affiche chaine de caracteres qui suit.
        DEFM  SPEED:
        DEFB  -1                  ;Signal: fin de chaine.
        LD    A,(VITESSE)
        SRL   A                   ;Divise par 2.
        SRL   A                   ;Par 4.
        SRL   A                   ;Par 8.
        DEC   A
        AND   31                  ;Modulo 32.
        LD    E,A                 ;Passe le resultat a E.
        LD    D,0                 ;Reset hi-byte.
        CALL  OUTNB2              ;Affiche le resultat.
        LD    BC,256+30           ;Ligne 1, colonne 1.
        CALL  LOCATE              ;Positionne le curseur.
        CALL  PRTL                ;Affiche chaine de caractere qui suit.
        DEFM  HISC:
        DEFB  -1                  ;Signal: fin de chaine.
        LD    DE,(HISCORE)
        JP    OUTNB               ;Affiche le hi-score.
;
;
AFTAB
;
; Affiche le tableau de Jeux.
;
        LD    A,2                 ;Active la fenetre 2.
        CALL  WINDOW
        CALL  CLW                 ;Efface la fenetre.
        CALL  CADRE               ;Affiche le cadre.
;
        LD    A,(NIVEAU)
        LD    B,A                 ;B sert de compteur.
        PUSH  BC                  ;Sauvegarde le niveau.
CHAMPL1
```

```
        PUSH BC                         ;Sauvegarde le compteur.
        CALL AFCHP                      ;Affiche 2 champignons
        CALL AFCHP                      ;  par niveau.
        POP  BC                         ;Retire le compteur.
        DJNZ CHAMPL1

        LD   (IX+NBFRUIT-AX),0
        LD   A,3                        ;Pen 3.
        CALL SETPEN
        POP  BC                         ;Retire le niveau de jeux.
        SLA  B                          ;Multiplie B par 2.
        SLA  B                          ;Par 4.
FRUITL1
        PUSH BC                         ;Sauvegarde le compteur.
        CALL AFRUIT                     ;Affiche un fruit.
        POP  BC                         ;Retire le compteur.
        DJNZ FRUITL1
        CALL KMRESET                    ;Vide le buffer d'entree du clavier.
        LD   A,1                        ;Pen 1.
        CALL SETPEN
        LD   BC,10*256+18               ;Ligne 10, colone 18.
        CALL LOCATE                     ;Positionne le curseur.
        LD   A,CHENI                    ;Affiche la tete de la chenille.
        JP   PRINT
;
CADRE
;
; Affiche le cadre de la fenetre 2.
;
        LD   A,2                        ;Pen 2.
        CALL SETPEN
        LD   BC,256+1                   ;Ligne 1, colonne 1.
        CALL LOCATE                     ;Positionne le curseur.
        LD   A,BORD                     ;Caractere de bordure.
        LD   B,40
CADREL1 CALL PRINT
        DJNZ CADREL1                    ;Affiche 40 fois.
;
        LD   BC,22*256+1                ;Ligne 22, colonne 1.
        CALL LOCATE
        LD   A,BORD
        LD   B,40
CADREL2 CALL PRINT
        DJNZ CADREL2                    ;Affiche 40 fois.
;
        LD   BC,2*256+1                 ;Ligne 2, colonne 1.
        CALL LOCATE
        LD   B,20
CADREL3 LD   A,BORD
        CALL PRINT
        LD   A,10                       ;Ligne suivante.
        CALL PRINT
        LD   A,8                        ;recule d'une case.
        CALL PRINT
        DJNZ CADREL3                    ;Affiche 20 fois.
;
        LD   BC,2*256+40                ;Ligne 2, colonne 40.
        CALL LOCATE
        LD   B,20
CADREL4 LD   A,BORD
        CALL PRINT
        LD   A,8                        ;Recule d'une case.
        CALL PRINT
        LD   A,10                       ;Ligne suivante.
```

```
        CALL PRINT
        DJNZ CADREL4                    ;Affiche 20 fois.
        RET
;
AFCHP
;
; Calcule une position et affiche un champignon.
;
AFCHPL1
        CALL RND                        ;Produit un nombre aleatoire dans A.
        AND  15                         ;Modulo 16.
        CP   10                         ;Modulo 9.
        JR   C,AFCHPP1
        SUB  7
AFCHPP1
        ADD  A,A                        ;Multiplie par 2.
        INC  A                          ;Produit un nombre pair
        INC  A                          ;  entre 2 et 18 inclus.
        LD   B,A                        ;Resultat dans B  (ligne).
        CALL RND                        ;Produit un nombre aleatoire dans A.
        AND  31                         ;Modulo 32.
        CP   20                         ;Modulo 20.
        JR   C,AFCHPP2
        SUB  14
AFCHPP2
        ADD  A,A                        ;Multiplie par 2.
        INC  A                          ;Produit un nombre pair
        INC  A                          ;  entre 2 et 38 inclus.
        LD   C,A                        ;Resultat dans C  (colonne).
        CALL TESTPOS                    ;La position est occupee ?
        JR   NC,AFCHPL1                 ;Si oui, calcule une autre position.
        CALL TESTMID                    ;Est-ce le milieu de la fenetre ?.
        JR   Z,AFCHPL1                  ;Si oui, recalcule une autre position.

        LD   A,CHAMPI
        CALL PRINT                      ;Affiche un quart du champignon.
        INC  A                          ;Affiche un autre quart.
        CALL PRINT
        INC  B                          ;Ligne suivante.
        PUSH AF
        CALL LOCATE                     ;Positionne curseur.
        POP  AF
        INC  A                          ;Affiche le troisieme quart.
        CALL PRINT
        INC  A                          ;Affiche le dernier quart.
        JP   PRINT
;
;
AFRUIT
;
; Calcule une position et affiche un fruit.
;
        CALL RND                        ;Produit un nombre aleatoire dans A.
        AND  31                         ;Modulo 32.
        CP   20                         ;Modulo 20.
        JR   C,AFRUITP1
        SUB  12
AFRUITP1
        INC  A
        INC  A                          ;Intervalle  2 a 21.
        LD   B,A                        ;Numero de ligne dans B.
        CALL RND                        ;Produit nombre aleatoire dans A.
        AND  63                         ;Modulo 64.
        CP   38                         ;Modulo 38.
```

```
         JR   C,AFRUITP2
         SUB  26
AFRUITP2
         INC  A
         INC  A                        ;Intervalle 2 a 39.
         LD   C,A                      ;Numero de colonne dans C.
         CALL TESTPOS                  ;Cette position est occupee ?
         JR   NC,AFRUIT                ;Si il y a un obstacle, re-calcule.
         JR   Z,AFRUIT                 ;Si il y a un fruit aussi.
         CALL TESTMID                  ;Est-ce le milieu de l'ecran ?
         JR   Z,AFRUIT                 ;Si oui, re-calcule.
         INC  (IX+NBFRUIT-AX)          ;Un fruit de plus.
         LD   A,FRUIT                  ;Affiche-le.
         JP   PRINT
;
READY
;
; Attend une touche et agit en consequence.
;
         LD   A,3                      ;Active la fenetre 3.
         CALL WINDOW
         CALL CLW                      ;Efface la fenetre.
         CALL PRTL                     ;Affiche chaine de caracteres qui suit.
         DEFM OK ? (o/n), v =
         DEFM vitesse, h = help
         DEFB -1                       ;Signal: fin de chaine.
         CALL WAITCHAR                 ;Attend un caractere du clavier.
         PUSH AF
         CALL CLW                      ;Avant de le traiter Efface la fenetre.
         POP  AF
         RES  5,A                      ;Converti les minuscules en majuscules.
         CALL READYT                   ;Traite le caractere.
         JR   NC,READY                 ;Si c'etait une commande alors recom.
         RET
READYT
         CP   "N"
         JR   Z,NOREADY                ;Re-affiche un tableau.
         CP   "V"
         JR   Z,SPEED                  ;Change la vitesse de jeux.
         CP   "H"
         JR   Z,AFHELP                 ;Affiche help.
         SCF                           ;Signal: pas de commande.
         RET
;
NOREADY
         CALL AFTAB                    ;Re-affiche un tableau.
         OR   A                        ;Signal: commande.
         RET
;
SPEED
         CALL PRTL                     ;Affiche chaine suivante.
         DEFM / ET \ POUR MOINS ET
         DEFM PLUS.
         DEFB -1                       ;Signal: fin de chaine.
SPEEDL1
         CALL WAITCHAR                 ;Attend un caractere.
         LD   C,(IX+VITESSE-AX)
         CP   "/"                      ;Moins vite ?
         JR   Z,SPEEDP1
         CP   "\"                      ;Plus vite ?
         JR   Z,SPEEDP2
         OR   A                        ;Signal: commande.
         RET
SPEEDP1
```

```
        LD   A,C
        SUB  8                      ;Reduit la vitesse.
        JR   SPEEDP3
SPEEDP2
        LD   A,C
        ADD  A,8                    ;Augmente la vitesse.
SPEEDP3 LD   (VITESSE),A
        CALL AFWIND1                ;Affiche la nouvelle vitesse.
        JR   SPEEDL1                ;Recommence.
;
AFHELP
        CALL HELP                   ;Affiche help.
        CALL AFTAB                  ;Affiche le tableau de jeux.
        OR   A                      ;Signal commande.
        RET
;
HELP
        LD   A,2                    ;Active la fenetre 2.
        CALL WINDOW
        CALL CLW                    ;Efface la fenetre.
        CALL CADRE                  ;Affiche le cadre.
        LD   A,1                    ;Pen 1.
        CALL SETPEN
        LD   BC,4*256+6             ;Ligne 4, colone 6.
        CALL LOCATE
        CALL PRTL
        DEFM Vous etes aux commandes d
        DEFM 'une
        DEFB 31,3,5
        DEFM chenille folle qui se nou
        DEFM rrit
        DEFB 31,3,6
        DEFM de fruits.
        DEFB 31,6,8
        DEFM Pour la guider vo
        DEFM us disposez des
        DEFB 31,3,9
        DEFM touches / et \ pour droi
        DEFM te et gauche.
        DEFB -1                     ;Signal: fin de chaine.
        JP   WAITCHAR               ;Attend un caractere du clavier.
;
;
TESTMID
;
; Teste si les coordonees BC sont celle du millieu de l'ecran.
;
        LD   A,B
        CP   10
        RET  NZ
        LD   A,C
        CP   18
        RET
;
;
RND
;
; Produit un nombre pseudo-aleatoire dans A.
;
        LD   HL,(SEED)              ;Ancien nombre aleatoire.
        PUSH HL
        ADD  HL,HL                  ;Multiplie par 2.
        ADD  HL,HL                  ;Par 4.
        POP  DE
```

```
        ADD   HL,DE                         ;Par 5.
        ADD   HL,DE                         ;Par 6.
        ADD   HL,DE                         ;Par 7.
        LD    DE,41
        ADD   HL,DE
        LD    (SEED),HL                     ;Stocke nouveau nombre.
        LD    A,L
        XOR   H                             ;Reduit  de 16 bits en 8 bits.
        RET
;
RANDOM
;
; Cree le Hasard.
;
        LD    A,R
        LD    (SEED),A
        RET
;
;
LOCATE
;
; Positionne le curseur.
;
; Entree  BC contient la ligne et la colonne.
;
        PUSH HL
        LD    A,B
        CALL SETROW                         ;Fixe ligne.
        LD    A,C
        CALL SETCOL                         ;Fixe colonne.
        POP   HL
        RET
;
;
INITP
;
; Initialisation
;
        LD    A,1                           ;Mode 1.
        CALL SETMODE
        CALL REDEFC                         ;Redefinit les caracteres.
        LD    A,1                           ;Fenetre 1.
        CALL WINDOW
        LD    HL,0                          ;Coin haut gauche.
        LD    DE,39*256+1                   ;Coin bas droite.
        CALL WINDEN                         ;Cree la fenetre.
        LD    A,2                           ;Fenetre 2.
        CALL WINDOW
        LD    HL,2                          ;Coin haut gauche.
        LD    DE,39*256+23                  ;Coin bas droite.
        CALL WINDEN                         ;Cree la fenetre.
        LD    A,3                           ;fenetre 3.
        CALL WINDOW
        LD    HL,24                         ;Coin haut gauche.
        LD    DE,39*256+24                  ;Coin bas droite.
        CALL WINDEN                         ;Cree la fenetre.
        LD    A,0                           ;Ink 0.
        LD    BC,INK0*256+INK0              ;Couleurs dans B et C.
        CALL SETINK
        LD    A,1                           ;Ink 1.
        LD    BC,INK1*256+INK1              ;Couleurs dans B et C.
        CALL SETINK
        LD    A,2                           ;Ink 2.
        LD    BC,INK2*256+INK2              ;Couleurs dans B et C.
```

```
        CALL SETINK
        LD   A,3                    ;Ink 3.
        LD   BC,INK3*256+INK3       ;Couleurs dans B et C.
        CALL SETINK
        LD   HL,AMPLENV             ;Adresse des donnees d'enveloppe.
        LD   A,1                    ;Enveloppe numero 1.
        CALL SDAMPLI                ;Cree l'enveloppe.
        CALL RANDOM                 ;Cree le hasard.
        LD   IX,AX                  ;IX pointe sur les variables.
        RET
;
;
REDEFC
;
; Redefini les caracteres.
;
        LD   C,240                  ;Premier caractere redefini.
        LD   HL,TABCC               ;Table des caracteres modeles.
        LD   B,8                    ;8 caracteres a redefinir.
REDEFCL1
        PUSH BC
        LD   A,25                   ;Signal: Redefinition.
        CALL PRINT
        LD   A,C                    ;Code du caractere a redefinir.
        CALL PRINT
        LD   B,8                    ;Compteur de parametres.
REDEFCL2
        LD   A,(HL)                 ;Prends un parametre.
        INC  HL
        CALL PRINT                  ;Envoie-le.
        DJNZ REDEFCL2               ;Recommence huit fois.
        POP  BC
        INC  C                      ;Caractere suivant.
        DJNZ REDEFCL1               ;Recommence 8 fois.
        RET
;
PRTL
;
; Affiche la chaine de caractere adressee par PC.
;
; Entree: PC = adresse de la chaine de caracteres.
;
        POP  DE                     ;Retire l'adresse.
        LD   A,(DE)                 ;Prend le caractere.
        INC  DE                     ;Pointe sur le caractere suivant
        PUSH DE                     ;  et stocke son adresse.
        CP   255                    ;Est-ce le caractere de fin ?
        RET  Z                      ;Si oui alors fin.
        CALL PRINT                  ;Si non affiche-le
        JR   PRTL                   ;  et continue.
;
;
OUTNB
;
; Affiche un nombre sur 5 chiffres.
;
; Entree: DE = nombre a afficher.
;
        LD   BC,-10000              ;Chiffre des dizaines de milliers.
        CALL PRCH                   ;Sort le chiffre.
        LD   BC,-1000               ;Chiffre des milliers.
        CALL PRCH
        LD   BC,-100                ;Chiffre des centaines.
        CALL PRCH
```

```
OUTNB2  LD   BC,-10              ;Chiffre des dizaines.
        CALL PRCH
        LD   A,E
        ADD  A,48                ;Chiffre des unites.
        JP   PRINT
;
PRCH
;
; Affiche un chiffre
;
; Entree: BC contient l'oppose du multiplicateur du chiffre.
;       : DE contient le nombre duquel on extrait le chiffre.
;
        EX   DE,HL               ;Passe le nombre a HL pour la division.
        XOR  A                   ;Compteur a 0.
PRCHL1  ADD  HL,BC               ;Soustrais le diviseur.
        INC  A                   ;Compte chaque passage.
        JR   C,PRCHL1            ;Tant que c'est possible.
        SBC  HL,BC               ;Restaure la soustraction de trop.
        EX   DE,HL               ;Le reste de la division dans DE.
        ADD  A,47                ;Resultat en code ASCII.
        JP   PRINT               ;  et affiche le chiffre.
;
;
SON
;
; Produit un son.
;
; Entree: HL = duree du son.
;       : DE = Frequence du son.
;          A = Numero d'enveloppe a utiliser.
;
        PUSH IX
        PUSH DE
        LD   IY,SDPARAM          ;IY pointe sur table de parametres.
        LD   (IY+1),A
        LD   (IY+7),L
        LD   (IY+8),H
        PUSH DE
        CALL SDRESET             ;Stoppe les sons.
        LD   (IY+0),%01000001    ;Signal: Canal 1.
        POP  DE
        CALL SETTONE             ;Inclue la frequence.
        CALL SOUND
        LD   (IY+0),%01000010    ;Signal: Canal 2.
        CALL EFFET               ;Desacorde pour faire un son puissant.
        CALL SOUND
        LD   (IY+0),%01000100    ;Signal: Canal 3.
        CALL EFFET
        CALL SOUND
        LD   A,7                 ;Signal: Tous les canaux.
        CALL SDRELEAS            ;Produit le son.
        POP  DE
        POP  IX
        RET
;
SOUND
        PUSH DE
        LD   HL,SDPARAM
        CALL SDQUEUE             ;Met le son dans la queue d'attente.
        POP  DE
        RET
;
EFFET
```

```
        LD   A,R
        AND  7                          ;Modulo8.
        INC  A
        EX   DE,HL
        LD   E,A
        LD   D,0
        ADD  HL,DE
        EX   DE,HL                      ;Nouvelle frequence dans DE.
SETTONE
        LD   (IY+3),E
        LD   (IY+4),D
        RET
;
SDPARAM
        DEFB 0                          ;Canal.
        DEFB 0                          ;Numero d'enveloppe.
        DEFB 0                          ;Enveloppe de frequence.
        DEFW 0                          ;Frequence du son.
        DEFB 0                          ;Periode de bruit.
        DEFB 0                          ;Amplitude initiale.
        DEFW 0                          ;Duree
;
AMPLENV
        DEFB 2                          ;Nombre de sections d'env. utilises.
;
        DEFB 1                          ;Nombre de pas.
        DEFB 15                         ;Taille du pas.
        DEFB 1                          ;Temps de pause.
;
        DEFB 15                         ;Nombre de pas.
        DEFB -1                         ;Taille du pas.
        DEFB 3                          ;Temps de pause.
;
        DEFS 9,0                        ;Parametres inutilises.
;
;
BUFFER
        DEFS LONGUEUR,0                 ;Place pour la chenille.
ENDBUFF
;
TABCC
;
CHENI   EQU  240
;
        DEFB %00111100
        DEFB %01011110
        DEFB %10011111
        DEFB %11111111
        DEFB %11111111
        DEFB %11111111
        DEFB %01111110
        DEFB %00111100
;
MORT    EQU  241
;
        DEFB %00111100
        DEFB %01100110
        DEFB %11000011
        DEFB %10000001
        DEFB %10000001
        DEFB %11000011
        DEFB %01100110
        DEFB %00111100
```

```
FRUIT     EQU  242
;
          DEFB %00000010
          DEFB %00000100
          DEFB %00001000
          DEFB %01010110
          DEFB %11111111
          DEFB %11111110
          DEFB %01111110
          DEFB %00101100
;
BORD      EQU  243
;
          DEFB %00011000
          DEFB %00011000
          DEFB %11111111
          DEFB %11111111
          DEFB %00011000
          DEFB %11111111
          DEFB %11111111
          DEFB %00011000
;
CHAMPI    EQU  244
;
          DEFB %00000000
          DEFB %00011111
          DEFB %01111111
          DEFB %11111111
          DEFB %11111111
          DEFB %11111100
          DEFB %01100011
          DEFB %00000111
;
          DEFB %00000000
          DEFB %11110000
          DEFB %11111100
          DEFB %11111110
          DEFB %11111111
          DEFB %01111110
          DEFB %10000000
          DEFB %11100000
;
          DEFB %00011111
          DEFB %00100111
          DEFB %00000111
          DEFB %00001111
          DEFB %00011111
          DEFB %00011111
          DEFB %00001111
          DEFB %00000011
;
          DEFB %11111000
          DEFB %11100100
          DEFB %11100000
          DEFB %11110000
          DEFB %11110000
          DEFB %11110000
          DEFB %11100000
          DEFB %11000000
;
FIN

Text:16300          End:34709           18409 Bytes
Hmem:36000
```

Liste des variables :

NIVEAU :	1 Octet contient le niveau actuel du jeu. Le premier tableau correspond au niveau 1.
TETE :	2 Octets cette variable contient l'adresse dans le buffer de la chenille des prochaines coordonnées ligne et colonne de la tête. Lorsque cette adresse dépasse la limite supérieure du buffer, elle est automatiquement remise au début de celle-ci.
QUEUE :	2 Octets contient l'adresse dans le buffer de la chenille des coordonnées ligne et colonnes de la queue.
DIGERE :	1 Octet contient le temps de digestion de la chenille.
DIRECT :	1 Octet contient le code de la direction de la chenille : 0 = à droite 1 = en bas 2 = à gauche 3 = en haut
SEED :	2 Octets stocke le dernier nombre aléatoire produit.
SCORE :	2 Octets contient le score.
HISCORE :	2 Octets contient le score le plus fort.
VITESSE :	1 Octet contient une valeur qui doit être un multiple de 8 et qui définit la vitesse du jeu.

NBFRUIT : 1 Octet contient le nombre de fruits qu'il reste à manger dans le tableau.

COORD : 2 Octets contient les coordonnées ligne et colonne de la position actuelle de la tête.

COMMENTAIRES DU LISTING

La première ligne utile du programme définit l'origine. Celle-ci est fixée à 37000 mais elle peut être changée.

Ensuite on définit les adresses des routines de la ROM qui sont utilisées. Elles sont toutes appelées par les vecteurs du système de manière à assurer la comptabilité avec les trois types d'ordinateur.
Ensuite viennent les constantes et les variables suivies du programme .

La touche [ESCAPE] vous permettra de sortir du jeu et de revenir à votre assembleur. Pour cela vous devez indiquer quelle est l'adresse d'accès de votre assembleur. Ceci se fait au début du listing dans la déclaration :

BREAK EQU x

x étant l'adresse d'accès.

Bon jeu !

10. Conclusion

Maintenant que vous avez étudié cet ouvrage de façon plus ou moins intensive, nous espérons que vous aurez mis en pratique un certain nombre des idées proposées.

Vous vous êtes peut-être aperçu qu'un certain nombre de sections de programme et de routines se répètent à travers les différents chapitres. Pensez simplement par exemple au menu avec ses fonctions:

-entrer

-examiner

-charger

-sauvegarder

-corriger

Bien sûr, pour des problèmes qui se ressemblent fondamentalement, il est possible de développer des normes de programmation qui standardisent le déroulement du programme. Nous y avons cependant renoncé pour rester plus proche de la façon de travailler et de réfléchir des programmeurs hobbyistes et amateurs. Ceux-ci partent en effet en général d'un sujet ou d'un problème et recherchent une solution sans avoir derrière la tête d'autres projets de logiciels. Le présent ouvrage constitue donc ainsi une aide pour le sous-programmeur débutant qui souhaite s'exercer et s'entraîner à la programmation.

ANNEXE A

Instructions BASIC importantes

Comme toutes les instructions BASIC AMSTRAD sont décrites en détail dans votre manuel d'utilisation, nous voulons simplement présenter ici quelques règles élémentaires pour leur utilisation:

1. Tout programme BASIC se termine par l'instruction END

2. Les expressions devant être sorties par une instruction PRINT doivent être séparées par des virgules si la sortie doit se poursuivre dans la prochaine zone de l'écran. Si vous utilisez le point-virgule, la sortie se fera immédiatement à la suite de la sortie déjà effectuée. Une virgule ou un point-virgule à la fin d'une instruction PRINT interdit le passage à la ligne qui est sinon systématique.

3. L'instruction INPUT réclame exactement autant de valeurs qu'il y a de variables dans la liste des variables. Les différentes valeurs ainsi que les différentes variables doivent être séparées entre elles par des virgules.

4. Si la condition dans l'instruction IF ... THEN est remplie, l'instruction suivant le THEN sera exécutée. D'autres instructions, séparées entre elles par des doubles points et placées à la suite de l'instruction IF ... THEN ne seront exécutées que si la condition logique est remplie. Si la condition n'est pas remplie, c'est tout simplement la ligne de programme suivante qui sera exécutée.

5. Dans l'instruction FOR ... TO ... NEXT (boucle) les valeurs initiale, finale et le pas de progression ne doivent pas être modifiés pendant les parcours de la boucle. Le nom de la variable de comptage doit être du type réel et il doit être identique au nom indiqué dans l'instruction NEXT. Les boucles peuvent être imbriquées, abandonnées et il est possible de sauter à l'intérieur d'une boucle.

6. Si on a besoin pour une variable indexée de plus de place que l'ordinateur n'en réserve automatiquement au moment de la définition d'une variable, il faut réserver la place nécessaire par dimensionnement (instruction DIM).

7. Lorsqu'on crée des variables réelles, il faut veiller à ce que leur nom ne comprenne pas plus de 40 caractères (alphanumériques), le premier caractère devant obligatoirement être une lettre. Les variables de chaîne de caractères doivent être dotées du signe $.

8. Si des données sont transmises à l'ordinateur avec les instructions DATA et READ, les types de valeurs doivent correspondre aux types de variable, la première valeur de l'instruction DATA étant affectée à la première variable de l'instruction READ, etc...

9. A la suite de l'instruction ON ... GOTO ou ON ... GOSUB, aucune autre instruction figurant sur la même ligne ne sera exécutée.

10. Les instructions WINDOW recouvrent les effets d'instructions WINDOW précédentes.

ANNEXE B

Messages d'erreur importants

ARRAY ALREADY DIMENSIONED : un tableau a été dimensionné deux fois.

CANNOT CONTINUE : tentative de continuer l'exécution d'un programme après un message d'erreur ou une modification du programme.

DATA EXHAUSTED : il y a plus d'instructions READ qu'il n'y a de données dans les instructions DATA.

DIVISION BY ZERO : la division par zéro est interdite.

LINE DOES NOT EXIST : appel d'une ligne inexistante.

NEXT MISSING : il manque l'instruction NEXT dans une boucle

OVERFLOW : le résultat d'un calcul est trop grand.

REDO FROM START : l'utilisateur a essayé d'entrer des données non numériques en réponse à une instruction INPUT dans laquelle le programmeur avait placé une variable de type numérique. Le programme peut continuer dès que l'utilisateur aura entré une donnée numérique.

SUBSCRIPT OUT OF RANGE : appel d'un élément d'un tableau dépassant le cadre fixé lors du dimensionnement (qu'il s'agisse d'un dimensionnement automatique ou d'un dimensionnement avec l'instruction DIM).

SYNTAX ERROR : instruction inexistante ou employée de façon incorrecte.

TYPE MISMATCH : des variables numériques ont été traités comme des chaînes de caractères ou inversement.

UNEXPECTED NEXT : il manque l'instruction FOR dans une boucle.

UNEXPECTED RETURN : sous-programme sans GOSUB.

EAUTÉS NOUVEAUTÉS NOUV

LIVRE DU LECTEUR DE DISQUETTE AMSTRAD CPC (Tome 10)

Tout sur la programmation et la gestion des données avec le floppy DDI-1 et le 664 ! Utile au débutant comme au programmeur en langage machine. Contient le listing du DOS commenté, un utilitaire qui ajoute les fichiers RELATIFS à l'AMDOS avec de nouvelles commandes BASIC, un MONITEUR disque et beaucoup d'autres programmes et astuces... Ce livre est indispensable à tous ceux qui utilisent un floppy ou un 664 AMSTRAD.

Réf. : ML127
Prix : 149 FF





LE NOUVEL ATARI ST

Ce livre décrit la superbe machine qu'est l'ATARI ST.
Architecture, interfaces, operating system, le bios, GEM, LOGO, le processeur 68000, sont quelques-uns des thèmes abordés. Ce livre doit être lu par tous ceux qui suivent de près le monde de la micro-informatique.

Réf. : ML125
Prix : 129 FF

LE NOUVEAU COMMODORE 128

Ce livre présente le nouveau Commodore 128. Vous y trouverez un aperçu complet des possibilités du successeur du célèbre "64" et une présentation détaillée des trois operating system. Le super nouveau BASIC Commodore 7.0 est décrit ainsi que la configuration de la mémoire, la page zéro et le nouveau et rapide lecteur de disquette 1571. Pour tous les Commodoristes !

Réf. : ML130
Prix : 129 FF



MICRO APPLICATION MICRO APPLICATION

LES LIVRES AMSTRAD

TRUCS ET ASTUCES POUR L'AMSTRAD CPC (Tome 1)

C'est le livre que tout utilisateur d'un CPC doit posséder. De nombreux domaines sont couverts (graphismes, fenêtres, langage machine) et des super programmes sont inclus dans ce best-seller (gestion de fichiers, éditeur de texte et de sons...).

Réf. : ML112
Prix : 149 FF





PROGRAMMES BASIC POUR LE CPC 464

ALIMENTEZ VOTRE CPC 464

Ce livre contient de super programmes, notamment un désassembleur, un éditeur graphique, un éditeur de texte... Tous les programmes sont prêts à être tapés et abondamment commentés.

Réf. : ML119
Prix : 129 FF

LE BASIC AU BOUT DES DOIGTS CPC 464

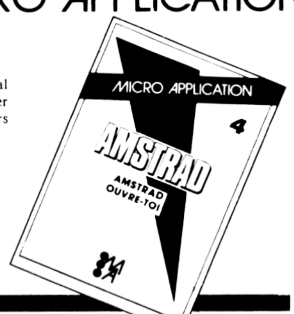
Ce livre est une introduction complète et didactique au BASIC du micro-ordinateur AMSTRAD CPC 464. Il permet d'apprendre rapidement et facilement la programmation (ins-tructions BASIC, analyses des problemes, algorythmes complexes...)
Principaux thèmes abordés :
- Les bases de la programmation
- Bit, Octet, ASCII
- Instructions du BASIC
- Organigrammes
- Les fenêtres
- Programmes BASIC plus poussés
- Le programme et menus.
Comprenant de nombreux exemples, ce livre vous assure un apprentissage simple et efficace du BASIC CPC 464.

Réf. : ML118
Prix : 149 FF



MICRO APPLICATION MICRO APPLICATION

MICRO APPLICATION MICRO APPLICATION



AMSTRAD OUVRE-TOI

Le bon départ avec le CPC 464 ! Ce livre vous apporte les principales informations sur l'utilisation, les possibilités de connexions du CPC 464 et les rudiments nécessaires pour développer vos propres programmes. C'est le livre idéal pour tous ceux qui veulent pénétrer dans l'univers des micro-ordinateurs avec le CPC 464.

Réf. : ML120
Prix : 99 FF

JEUX D'AVENTURES. COMMENT LES PROGRAMMER

Voici la clé du monde de l'aventure. Ce livre fournit un système d'aventures complet, avec éditeur, interpréteur, routines utilitaires et fichiers de jeux. Ainsi qu'un générateur d'aventures pour programmer vous-mêmes facilement vos jeux d'aventures. Avec, bien sûr, des programmes tout prêts à être tapés.

Réf. : ML121
Prix : 129 FF



LA BIBLE DU PROGRAMMEUR DE L'AMSTRAD CPC 464 (Tome 6)

Tout, absolument tout sur le CPC 464. Ce livre est l'ouvrage de référence pour tous ceux qui veulent programmer en pro leur CPC. Organisation de la mémoire, le contrôleur vidéo, les interfaces, l'interpréteur et toute la ROM DESASSEMBLEE et COMMENTEE sont quelques-uns des thèmes de cet ouvrage de 700 pages.

Réf. : ML122
Prix : 249 FF



LE LANGAGE MACHINE DE L'AMSTRAD CPC (Tome 7)

Ce livre est destiné à tous ceux qui désirent aller plus loin que le BASIC. Des bases de la programmation en assembleur à l'utilisation des routines système, tout est expliqué avec de nombreux exemples. Contient un programme assembleur, moniteur et désassembleur.

Réf. : ML123
Prix : 129 FF

MICRO APPLICATION MICRO APPLICATION

GRAPHISMES ET SONS DU CPC

L'AMSTRAD CPC dispose de capacités graphiques et sonores exceptionnelles. Ce livre en montre l'utilisation à l'aide de nombreux programmes utilitaires.
Contenu :
- base de programmation graphique
- éditeur de police de caractères
- "sprites", "shapes", et chaines
- représentations multi-couleurs.
- calcul des coordonnées
- rotations, mouvements
- représentations graphiques de fonctions en 3D
- D.A.O. (dessin assisté par ordinateur)
- synthétiseur
- mini-orgue
- enveloppes de son, et beaucoup d'autres choses...

Réf. : ML124
Prix : 129 FF





PEEKS ET POKES DU CPC (Tome 9)

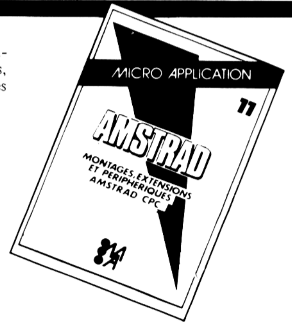
Comment exploiter à fond son CPC à partir du BASIC ? C'est ce que vous révèle ce livre avec tout ce qu'il faut savoir sur les peeks, pokes et autres call... Vous saurez aussi comment protéger la mémoire, calculer en binaire... et tout cela très facilement. Un passage, assuré et sans douleur du BASIC au puissant LANGAGE MACHINE.

Réf. : ML126
Prix : 99 FF

MONTAGES, EXTENSIONS ET PERIPHERIQUES AMSTRAD CPC (Tome 11)

Pour tous les amateurs d'éléctronique ce livre montre ce que l'on peut réaliser avec un CPC. De nombreux schémas et exemples illustrent les thèmes et applications abordés comme les interfaces, programmateur d'EPROM... Un très beau livre de 450 pages.

Réf. : ML131
Prix : 199 FF



LE LIVRE DU CP/M AMSTRAD
(Tome 12)

Ce livre vous permettra d'utiliser CP/M sur les CPC 464, 664 et 6128 sans aucune difficulté. Vous y trouverez de nombreuses explications et les différents exemples vous assureront une maitrise parfaite de ce très puissant système d'exploitation qu'est CP/M. (300 pages).

Réf. : ML128
Prix : 149 FF





DES IDEES POUR LES CPC
(Tome 13)

Vous n'avez pas d'idées pour utiliser votre CPC (464, 664, 6128) ? Ce livre va vous en donner ! Vous trouverez de très nombreux programmes BASIC couvrant des sujets très variés qui transformeront votre CPC en un bon petit génie. De plus les programmes vous permettront d'appronfondir vos connaissances en programmation.
(250 pages).

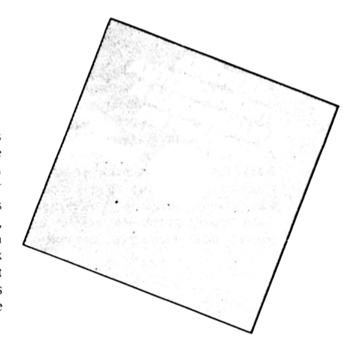
Réf. : ML132
Prix : 129 FF

AMSTRAD AUTOFORMATION
A L'ASSEMBLEUR
EN FRANCAIS

Contient un livre et un logiciel.
LE LIVRE :
Cet ouvrage introduit le débutant à la programmation du Z80 grâce à la méthode du DR WATSON qui selon les critiques vaut son pesant d'or ! Aucune connaissance préalable n'est requise et le but du livre est d'assurer au novice un succès total. A la fin du livre les instructions du Z80 sont expliquées en détail. De nombreux exemples illustrent les différentes étapes du cours alors que des exercices (les solutions sont fournies) testent la compréhension.
LE LOGICIEL : Un assembleur Z80 complet est livré sur cassette et comprend :
- Etiquettes Symboliques
- Directives d'Assemblage
- Chargement/Sauvegarde
- Copie Ecran
- INSERT / DELET.

L'assembleur permet d'écrire des programmes facilement en langage d'assemblage puis les transforme en code machine (langage machine). Pour vous aider à comprendre les rotations mathématiques utilisées, une démonstration de l'utilisation des nombres binaires et hexadecimaux est fournie. Un programme utilisant les commandes graphiques additionnelles décrites dans le livre est également fourni.

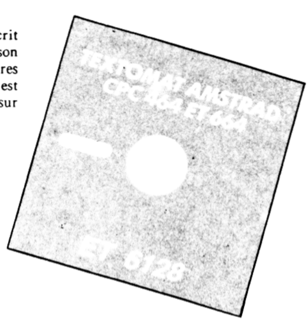
Réf. : ML126
Prix : 195 FF K 7 - 295 FF - disquette

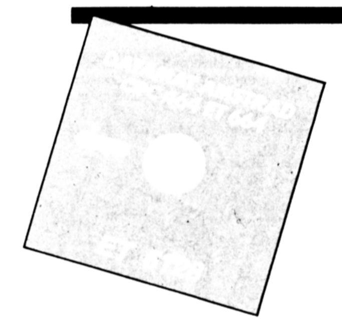


TEXTOMAT AMSTRAD
CPC 464 & 664

Traitement de texte de qualité professionnelle pour tous.
Tabulation, recherche, remplacement, insertion, manipulation de paragraphes, calcul... Accents à l'écran et imprimante. Module permettant de gérer tout type d'imprimante. Ecrit en LANGAGE MACHINE. Liaison avec DATAMAT pour mailing et lettres types personnalisées... TEXTOMAT est la solution traitement de texte sur CPC. Documentation complète.

Réf. : AM305
Prix : 450 FF





DATAMAT AMSTRAD
CPC 464 & 664

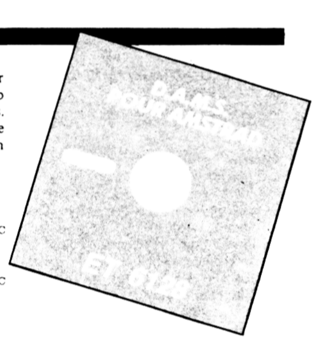
La gestion de fichier la plus complète fonctionnant pour les 464 et 664. Entièrement en LANGAGE MACHINE. Fonctions de calcul, de tri, de recherche multicritères, impressions paramètrables, liaison avec TEXTOMAT pour mailing... Documentation française de 60 pages.

Réf. : AM304
Prix : 450 FF

D.A.M.S. POUR AMSTRAD
CPC 464 & 664

D.A.M.S. est un logiciel intégrant un assembleur, un moniteur et un désassembleur symbolique pour développer et mettre au point facilement des programmes en langage machine sur les micro ordinateurs AMSTRAD. Les trois modules sont co-résidents en mémoire ce qui assure une grande souplesse d'utilisation. Vous pouvez notamment utiliser un éditeur plein écran, un assembleur immédiat, un désassembleur symbolique, une trace et beaucoup d'autres fonctions très puissantes. D.A.M.S. est entièrement relogeable et est bien évidemment écrit en langage machine.

Réf. : AM208
Prix : sur cassette : 295 FF TTC pour CPC 464
Réf. : AM308
Prix : sur disquette : 395 FF TTC pour CPC 664 & CPC 464



CE LIVRE VOUS PASSIONNE,

MAIS VOUS N'AVEZ PAS LE TEMPS

DE TAPER LES PROGRAMMES

ACHETEZ LA DISQUETTE

DES PROGRAMMES DU LIVRE

DES IDÉES POUR LES CPC

NOM : ________________________________

PRENOM : ________________________________

ADRESSE : ________________________________

____

CODE POSTAL : __________ **VILLE :** ____________________

Je désire recevoir la disquette des programmes du livre DES IDÉES POUR LES CPC.

Je joins a ce coupon un chèque de 120 francs.
CCP de 120 francs.

Date : **Signature :**

Achevé d'imprimer en décembre 1985
sur les presses de l'imprimerie Laballery et C[ie]
58500 Clamecy
Dépôt légal : décembre 1985
Numéro d'imprimeur : 511094

IMPRIMÉ EN FRANCE

[redacted]

Vous n'avez pas d'idées pour utiliser votre CPC (464, 664, 6128)? Ce livre va vous en donner! Vous trouverez de très nombreux programmes BASIC couvrant des sujets très variés qui transformeront votre CPC en un bon petit génie. De plus les programmes vous permettront d'approfondir vos connaissances en programmation.

Aperçu du contenu du livre :

1. Le CPC et ce qu'il permet de faire
2. De petites choses utiles et faciles à programmer
   — Calcul de pourcentage
   — Minimum, maximum et moyenne
   — L'ordre alphabétique
   — Jouez-vous au Loto?
   — Publicité dans une vitrine
3. Voiture et ordinateur
   — consommation d'essence — contrôle sur une longue période
   — Optimisation de l'itinéraire choisi
   — Coûts de la voiture
4. L'argent et le crédit
   — Calcul d'intérêts
   — L'argent de votre compte d'épargne (calcul d'intérêts des intérêts)
   — L'intérêt annuel effectif
   — Les dettes et leur remboursement
5. Textes et impression
   — Caractères spéciaux français
   — Une lettre d'affaire individuelle
   — Traitement de texte
6. Budget et santé
   — Un budget rationnel
   — Un calculateur de calories électronique
7. L'ordinateur et vos hobbies
   — Peinture, dessin et graphiques
   — Faisóns de la musique
   — Le championnat de football
8. Ces chers petits
   — La table de multiplication
   — Test de nombres premiers
   — Détermination de point zéro
   — Vocabulaire

Prix : 129 FF TTC
Réf. : ML 132

ISBN : 2-86899-031-2


AMSTRAD
DES IDÉES POUR LES CPC
13



Document numérisé avec amour par :

AMSTRAD CPC

MÉMOIRE ÉCRITE

ACME

https://acpc.me/